Technische Universität Ilmenau
Institut für Mathematik

Preprint No. M 01/12

Zur Numerik nichtlinearer Gleichungssysteme (Teil 1)

Vogt, Werner

Oktober 2001

Impressum:
Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69 3621
Fax: +49 3677 69 3270
http://www.tu-ilmenau.de/ifm/

ISSN xxxx-xxxx

ilmedia

Zur Numerik nichtlinearer Gleichungssysteme (Teil 1)

Werner Vogt
Technische Universität Ilmenau
Institut für Mathematik
Postfach 100565
98684 Ilmenau

Ilmenau, den 25.10.2001

Zusammenfassung Zwei grundlegende analytisch-numerische Zugänge zur Lösung nichtlinearer endlicher Gleichungssysteme – das Fixpunktprinzip und das Linearisierungsprinzip – werden vorgestellt, theoretisch begründet und algorithmisch aufbereitet. Nach Anwendung der Picard-Iteration, auch auf den Spezialfall linearer Systeme, wird das Newton-Verfahren nebst zweier Varianten betrachtet. Wesentliche Begriffe, wie Konvergenzordnung, Konvergenzbedingungen, a-priori- und a-posteriori-Fehlerschätzungen werden eingeführt und veranschaulicht.

1 Fixpunktiteration für nichtlineare Systeme

In zahlreichen mathematischen Modellen tritt als zentrales Problem die Lösung nichtlinearer endlichdimensionaler Gleichungssysteme auf. Eine oft anzutreffende Standardaufgabe kann durch das reelle System

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ \ldots\ldots\ldots\ldots\ldots & \quad . \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{1}$$

mit n Gleichungen für die n reellen Variablen (Unbekannten) $x_1, x_2, \ldots, x_n$ beschrieben werden. Faßt man die Variablen zum Spaltenvektor $x = (x_1, x_2, \ldots, x_n)^T$ zusammen und definiert den Funktionenvektor

$$f(x) = (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n))^T ,$$

so läßt sich dieses Gleichungssystem in der kompakteren Vektorschreibweise

$$f(x) \;=\; 0 \qquad \text{mit} \quad f:\; D \subset \mathbb{R}^n \to \;\mathbb{R}^n\,,\; D \text{ offen} \tag{2}$$

mit dem nichtleeren Definitionsbereich D notieren. Gesucht sind nun Vektoren $x^* \in D$, für die $f(x^*) = 0$ gilt. Derartige Lösungen der Aufgabe (2) werden nachfolgend auch als *Nullstellen* der Funktion f bezeichnet.

Beispiel 1.1 Typische Nullstellenaufgaben mit nichtlinearen Gleichungen sind

$$\begin{aligned} 4x_1 - x_2 - x_1 \sin x_2 &= 0 \\ (x_1 + x_2) \tan x_2 &= 0 \end{aligned} \tag{3}$$

oder die eindimensionale Aufgabe

$$\tan x - x \;=\; 0\,. \tag{4}$$

Außer transzendenten Gleichungen (d.h. Gleichungen, die transzendente Funktionen enthalten) gehören hierzu auch Systeme algebraischer Gleichungen

$$\begin{aligned} 2x_1^3 - x_2^2 - 1 &= 0 \\ x_1 x_2^3 - x_2 - 4 &= 0\,, \end{aligned} \tag{5}$$

für die reelle und mitunter auch komplexe Lösungen gesucht sind.

Großdimensionale Systeme entstehen bei der Diskretisierung von Differential- und Integralgleichungen; z.B. das System

$$\begin{array}{lcll} 2x_1 - x_2 & = & \lambda x_1(1-x_1) & \\ -x_{j-1} + 2x_j - x_{j+1} & = & \lambda x_j(1-x_j), & j = 2(1)n-1 \\ -x_{n-1} + 2x_n & = & \lambda x_n(1-x_n) & \end{array} \tag{6}$$

mit dem Parameter $\lambda > 0$. Um Lösungen für möglichst große Dimensionen n zu erhalten, werden angepaßte Verfahren angewendet, die die spezielle schwachbesetzte Struktur der Gleichungen ausnutzen.

Da in praktischen Anwendungen häufig Parameter auftreten, ist man besonders daran interessiert, die Abhängigkeit der Lösungen von einzelnen Systemparametern zu untersuchen. Ein Modell von W. F. Langford (1984) führt auf das Gleichungssystem

$$\begin{array}{lcl} (x_3 - 0.7)\cdot x_1 - \omega \cdot x_2 & = & 0 \\ \omega x_1 + (x_3 - 0.7)\cdot x_2 & = & 0 \\ 0.6 + x_3 - \frac{1}{3}x_3^3 - (x_1^2 + x_2^2)(1 + \rho \cdot x_3) + \varepsilon \cdot x_3 \cdot x_1^3 & = & 0 \end{array} \tag{7}$$

mit positiven reellen Parametern ω, ρ und ε .

Natürlich gehören auch lineare normalbestimmte Gleichungssysteme

$$Ax = a\,, \quad A \in \mathbb{R}^{n\times n}, \quad a \in \mathbb{R} \tag{8}$$

zur Aufgabenklasse (2). Viele Verfahren und deren theoretische Absicherung vereinfachen sich in diesem speziellen Fall und sind auch bei großdimensionalen linearen Aufgaben erfolgreich einsetzbar. ◀

Im Gegensatz zu linearen Systemen (8) mit einer geschlossenen Lösungstheorie kann im nichtlinearen Fall die Frage nach der Existenz und Eindeutigkeit von Lösungen nicht allgemein beantwortet werden. Existenzsätze sind im allgemeinen von lokaler Art, und die Anzahl und Lage der Lösungen variiert häufig bei Änderung der Systemparameter. Deshalb entscheidet die Wahl einer „geeigneten Startnäherung“ meist über Erfolg oder Mißerfolg der eingesetzten numerischen Verfahren.

Zudem können – abgesehen von algebraischen Gleichungen niederen Grades und speziellen transzendenten Systemen – die gesuchten Lösungen nicht in geschlossener Form angegeben werden. Damit sind alle vorgestellten Methoden *iterative Näherungsverfahren*, d.h. jede gesuchte Lösung x^* wird im allgemeinen nach endlich vielen Schritten nur angenähert bestimmt. Konvergenzuntersuchungen und effiziente Fehlerabschätzungen sind deshalb unabdingbar, um die erhaltenen Näherungslösungen bewerten zu können.

1.1 Die Fixpunktiteration (Picard-Verfahren)

Es ist bekannt, daß der Raum $\mathbb{R}^n$, versehen mit einer Norm, ein Banachraum ist. Die bekanntesten Normen sind

$$\|x\|_1 \;=\; \sum_{j=1}^{n} |x_j| \qquad \text{(Betragssummennorm)}$$

$$\begin{aligned} \|x\|_2 &= \sqrt{\sum_{j=1}^{n} |x_j|^2} \qquad \text{(Euklidische Norm)} \\ \|x\|_\infty &= \max_{j=1(1)n} |x_j| \qquad \text{(Betragsmaximumnorm).} \end{aligned}$$

Damit läßt sich der Banachsche Fixpunktsatz anwenden, nachdem das Nullstellenproblem (2) in die *Fixpunktform (iterierfähige Form)*

$$x = g(x) \ , \quad g : \mathbb{R}^n \to \mathbb{R}^n \tag{9}$$

überführt worden ist. Dabei ist $g: \ D \subset \mathbb{R}^n \to \ \mathbb{R}^n$ eine Vektorfunktion mit

$$g(x) := \begin{pmatrix} g_1(x_1, x_2, \ldots, x_n) \\ g_2(x_1, x_2, \ldots, x_n) \\ \cdots\cdots\cdots \\ g_n(x_1, x_2, \ldots, x_n) \end{pmatrix} .$$

Jede Lösung $x^* \in D$ der Gleichung (9) heißt *Fixpunkt* der Abbildung g.

Beispiel 1.2 Das Nullstellenproblem (3)

$$\begin{aligned} f_1(x_1, x_2) &= 4x_1 - x_2 - x_1 \sin x_2 = 0 \\ f_2(x_1, x_2) &= (x_1 + x_2) \tan x_2 \quad = 0 \end{aligned}$$

läßt sich in die Fixpunktform $x = g(x)$ mit der Abbildung

$$g(x) = \begin{pmatrix} \frac{1}{4}(x_1 \sin x_2 + x_2) \\ \arctan \frac{4}{x_1 + x_2} \end{pmatrix}$$

überführen. ◀

Die Transformation eines Nullstellenproblems in die Fixpunktform ist stets auf beliebig viele Weisen möglich. Dabei sollte jedoch auf die Lösungsäquivalenz geachtet werden, d.h. die Nullstellenmenge und die entsprechende Fixpunktmenge eines Problems sollten übereinstimmen.

Einen allgemein anwendbaren Weg garantiert folgende Äquivalenzaussage, die zugleich den Übergang von der Nullstellenform zur Fixpunktform beschreibt:

Satz 1.1 *Ist x^* Fixpunkt von g , so ist x^* Nullstelle von $f(x) := x - g(x)$. Sei andererseits $\Phi : D \to \mathbb{R}^{n \times n}$ eine beliebige Matrixfunktion, die für jedes $x \in D$ regulär ist. Dann ist jede Nullstelle von f auch ein Fixpunkt von*

$$g(x) := x - \Phi(x) f(x) \, .$$

Die *Fixpunktiteration (allgemeines Iterationsverfahren, Picard-Verfahren)* besitzt die Form

$$\boxed{x_{k+1} = g(x_k) \ , \quad k = 0, 1, 2, 3, ...,} \tag{10}$$

mit Startnäherung $x_0 \in D \subset \mathbb{R}^n$. Das Verfahren *konvergiert* gegen einen Fixpunkt x^* von g, falls

$$\lim_{k\to\infty} \|x_k - x^*\| \;=\; 0$$

mit einer Norm des $\mathbb{R}^n$ gilt. Man schreibt dann $\lim_{k\to\infty} x_k = x^*$. Um die Konvergenz des Verfahrens (10) nachzuweisen, kann eine beliebige Norm gewählt werden, denn in $\mathbb{R}^n$ sind alle Normen äquivalent.

Lemma 1.1 *Alle Normen $||x||$ in $\mathbb{R}^n$ sind äquivalent in folgendem Sinne: Für jedes Paar von Normen $N_1(x)$ und $N_2(x)$ existieren Konstanten m und M, so daß*

$$m \cdot N_2(x) \le N_1(x) \le M \cdot N_2(x) \qquad \forall x \in \mathbb{R}^n$$

gilt.

Beweis: Sei speziell $N_2(x) := \|x\|_\infty = \max\limits_{i=1(1)n} |x_i|$ und $N_1(x)$ eine beliebige andere Norm. Die Menge $S = \{x \in \mathbb{R}^n |\;\; \|x\|_\infty = 1\}$ ist kompakt in $\mathbb{R}^n$. Wegen der Stetigkeit der Norm $N_1(x)$ existieren Konstanten

$$M := \max_{x\in S} N_1(x) > 0 \quad \text{und} \quad m := \min_{x\in S} N_1(x) > 0\,.$$

Sei $y \in \mathbb{R}^n, y \neq 0$, beliebig. Offenbar ist dann $y/\|y\|_\infty \in S$. Damit ist

$$m \le N_1\left(\frac{y}{\|y\|_\infty}\right) \le M,$$

woraus wegen der Normeigenschaft von N_1

$$m \cdot \|y\|_\infty \;\le\; N_1(y) \le M \cdot \|y\|_\infty$$

folgt. Trivialerweise gilt dies auch für $y = 0$, womit die Behauptung im Spezialfall $\|y\|_\infty = N_2(y)$ bewiesen ist. Für zwei beliebige Normen N_1, N_2 folgt nun für alle $x \in \mathbb{R}^n$

$$m_1\|x\|_\infty \le N_1(x) \le M_1 \cdot \|x\|_\infty\,,$$

$$m_2\|x\|_\infty \le N_2(x) \le M_2 \cdot \|x\|_\infty\,.$$

Man erhält damit die Abschätzungen

$$N_1(x) \le M_1 \cdot \|x\|_\infty \le \tfrac{M_1}{m_2} N_2 x = M \cdot N_2(x)\,,$$

$$N_1(x) \ge m_1 \cdot \|x\|_\infty \ge \tfrac{m_1}{M_2} N_2 x = m \cdot N_2(x)\,,$$

womit die Behauptung bewiesen wurde. □

Der zugehörige Algorithmus `picard` erfordert als Input die Funktion g, einen Startwert x und die (absolute und relative) Toleranz $tolabs, tolrel$.

Algorithmus 1.1 (Picard-Verfahren)

Function picard$(g, x, tolabs, tolrel)$

1. Berechne Toleranz $tol = tolrel \cdot ||x|| + tolabs$ und $y = g(x)$
2. Do while $||y - x|| > tol$
   1. Überspeichere $x = y$
   2. Berechne $y = g(x)$
3. Return y

Wendet man den Banachschen Fixpunktsatz auf das Verfahren (10) an, so hat man außer der Kontraktionsbedingung zu verifizieren, daß die Funktion g eine abgeschlossene Menge M in sich abbildet. Gerade der Nachweis dieser zweiten Eigenschaft bietet im mehrdimensionalen Fall oft beträchtliche Schwierigkeiten. Eine Variante, bei der eine Bedingung an die Startlösung x_0 zu erfüllen ist, stellt der folgende Satz dar.

Satz 1.2
Die Vektorfunktion g sei zu gegebenem x^0 auf einer Kugel $S_0 = \{x| \, \|x - x^0\| \leq \varrho\}$ definiert und Lipschitz-stetig mit

$$\|g(x) - g(y)\| \;\leq\; \lambda \|x - y\| \quad \forall\, x, y \in S \;,$$

wobei $0 \leq \lambda < 1$ ist (Kontraktionsbedingung). *Der Startvektor x_0 genüge zudem der Ungleichung*

$$\|g(x_0) - x_0\| \;\leq\; (1 - \lambda)\varrho \;.$$

Dann gelten folgende Behauptungen:

(i) *Alle Iterationsvektoren x_k des Verfahrens $x_{k+1} = g(x_k)$ verbleiben in S.*

(ii) *Die x_k konvergieren gegen einen Vektor x^*, der eine Lösung von $x = g(x)$ ist.*

(iii) *Der Vektor x^* ist der einzige Fixpunkt in der Kugel S.*

(iv) *Für den k-ten Iterationsvektor x_k gilt die a-priori–Fehlerschätzung*

$$\|x_k - x^*\| \leq \frac{\lambda^n}{1 - \lambda} \|x_1 - x_0\|$$

sowie die a-posteriori–Fehlerschätzung

$$\|x_k - x^*\| \leq \frac{\lambda}{1 - \lambda} \|x_k - x_{k-1}\| \;.$$

BEWEIS: Man zeigt Behauptung (i) induktiv. Für $k = 1$ gilt nach Voraussetzung $\|g(x_0) - x_0\| \leq (1-\lambda)\varrho \leq \varrho$, also ist $x_1 \in S$. Wenn bereits $x_1, x_2, \ldots, x_k \in S$ erfüllt ist, so liefert die Kontraktionsbedingung

$$\begin{aligned} ||x_{k+1} - x_k|| &= \|g(x_k) - g(x_{k-1})\| \leq \lambda \|x_k - x_{k-1}\| \\ &\leq \lambda^2 \|x_{k-1} - x_{k-2}\| \leq \ldots \leq \lambda^k \|x_1 - x_0\| \\ &\leq \lambda^k (1-\lambda)\varrho \,. \end{aligned}$$

Mit dieser Abschätzung und der Dreiecksungleichung erhält man daraus nach Summation der entstehenden geometrischen Reihe

$$\begin{aligned} ||x_{k+1} - x_0|| &\leq \|x_{k+1} - x_k\| + \|x_k - x_{k-1}\| + \ldots + \|x_1 - x_0\| \\ &\leq (\lambda^k + \lambda^{k-1} + \ldots + \lambda + 1)(1-\lambda)\varrho \\ &\leq (1 - \lambda^{k+1})\varrho \leq \varrho \,, \end{aligned}$$

also ist auch $x_{k+1} \in S$. Um Behauptung (ii) nachzuweisen, betrachtet man zwei Indizes k und m mit $k > m$. Analog zu (i) schätzt man dann ab

$$\begin{aligned} ||x_k - x_m|| &\leq \|x_k - x_{k-1}\| + \|x_{k-1} - x_{k-2}\| + \ldots + \|x_{m+1} - x_m\| \\ &\leq (\lambda^{k-1} + \lambda^{k-2} + \ldots + \lambda^m)||x_1 - x_0|| \\ &\leq (\lambda^{k-1-m} + \lambda^{k-2-m} + \ldots + \lambda + 1)\lambda^m (1 - \lambda^{k+1})\varrho \\ &\leq \lambda^m \varrho \,, \end{aligned}$$

womit $||x_k - x_m|| \to 0$ für $k, m \to \infty$ folgt. Also ist $\{x_k\}$ eine Cauchyfolge und besitzt einen Grenzwert $\xi = \lim_{k\to\infty} x_k$. Wegen $x_k \in S$ ist dann auch $\xi \in S$. Wegen der Lipschitz-Stetigkeit ist offenbar $\|g(x_k) - g(\xi)\| \leq \lambda \|x_k - \xi\|$, also $\lim_{k\to\infty} g(x_k) = g(\xi)$, woraus die Eigenschaft

$$\xi = \lim_{k to \infty} x_{k+1} = \lim_{k to \infty} g(x_k) = g(\xi)$$

folgt. Damit ist $x^* = \xi$ ein Fixpunkt von g. Die Eindeutigkeit des Fixpunktes zeigt man indirekt: Gäbe es nämlich einen weiteren Fixpunkt $x^{**} \in S, x^{**} \neq x^*$, so folgte mit der Kontraktionsbedingung

$$||x^{**} - x^*|| = \|g(x^{**}) - g(x^*)\| \leq \lambda ||x^{**} - x^*|| < ||x^{**} - x^*|| \,,$$

also ein Widerspruch. Die Fehlerschätzungen (iv) erhält man völlig analog zu Satz 1.1. □

Der Nachweis der Kontraktivität von g mittels der Lipschitz-Stetigkeit ist häufig aufwendig und nicht praktikabel. Wenn jedoch vorausgesezt werden kann, daß die Funktionen $g_i(x_1, x_2, \ldots, x_n)$ stetig sind und stetige partielle Ableitungen besitzen, so können einfachere Kriterien für die Kontraktivität angewendet werden. Dazu werden jedoch Eigenschaften von Matrixnormen benötigt.

1.2 Matrixnormen und Spektralradius

Wir betrachten den reellen n-dimensionalen Vektorraum $\mathbb{R}^n$ mit einer Norm $||.||$. Durch jede reelle (n, n)-Matrix A wird eine eindeutige Abbildung von $\mathbb{R}^n$ in $\mathbb{R}^n$ der Form $y = Ax$ vermittelt. Diese Abbildung ist linear, denn es gilt:

$$A(\alpha x + \beta y) = \alpha Ax + \beta Ay \quad \text{für alle } x, y \in \mathbb{R}^n \,.$$

Die Abbildung $A|x \to y$ heißt *beschränkt*, wenn eine Konstante $C > 0$ existiert, so daß für alle $x \in \mathbb{R}^n$

$$\|Ax\| \leq C \cdot \|x\| \tag{11}$$

ist. Die kleinste derartige Schranke (also das Infimum der Menge aller Schranken) ist eine charakteristische Zahl für die Matrix A.

Definition 1.1 (Matrixnorm) *Die kleinste Zahl C, für die $\|Ax\| \leq C\|x\| \quad \forall x \in \mathbb{R}^n$ gilt, heißt* Norm der Matrix *A und wird mit $\|A\|$ bezeichnet. Sie wird durch die gegebene Vektornorm* induziert.

Offenbar ist die induzierte Matrixnorm genau die aus der Funktionalanalysis bekannte Norm für den entsprechenden linearen Operator. Damit läßt sie sich äquivalent formulieren:

Lemma 1.2

$$\|A\| \;=\; \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \;=\; \max_{||x||=1} \|Ax\|$$

Beweis: Wir zeigen zuerst die linke Gleichheit. Sei $\alpha := \sup_{x \neq 0}(\|Ax\|/\|x\|)$. Offenbar ist dann $\|Ax\|/\|x\| \leq \alpha$ für alle $x \in \mathbb{R}^n,\ x \neq 0$, also $\|Ax\| \leq \alpha\, \|x\|$ für alle $x \in \mathbb{R}^n$. Nach Definition von $\|A\|$ ist damit $\alpha \geq \|A\|$.
Aus der Supremumeigenschaft folgt, daß für beliebiges $\varepsilon > 0$ ein Element $x_\varepsilon \neq 0$ existiert, so daß $\alpha - \varepsilon \leq \|Ax_\varepsilon\|/\|x_\varepsilon\|$ gilt bzw.

$$(\alpha - \varepsilon)\|x_\varepsilon\| \leq \|Ax_\varepsilon\| \leq C\|x_\varepsilon\| \,.$$

Folglich ist $\alpha - \varepsilon \leq C$ für alle Schranken C, also $\alpha - \varepsilon \leq inf\ C = \|A\|$. Da $\varepsilon > 0$ beliebig war, ist $\alpha \leq \|A\|$. Aus beiden Ungleichungen folgt die linke Gleichheit des Lemmas. Mit den Normeigenschaften erhält man schließlich

$$\sup_{y \neq 0} \frac{\|Ay\|}{\|y\|} \;=\; \sup_{y \neq 0} \|A \frac{y}{||y||}\| \;=\; \max_{||x||=1} \|Ax\| \,,$$

womit das Lemma bewiesen wurde. □

Jede durch eine Vektornorm $\|.\|$ induzierte Matrixnorm besitzt folgende leicht nachweisbare *Normeigenschaften*:

(i) $\|A\| \geq 0\,, \quad \|A\| = 0 \;\Leftrightarrow\; A = O$ (Nullmatrix)
(ii) $\|\lambda A\| = |\lambda|\|A\|\,, \quad \forall \lambda \in \mathbb{R}\,.$
(iii) $\|A + B\| \leq \|A\| + \|B\|$.
(iv) $\|AB\| \leq \|A\| \cdot \|B\|$.

Aus Eigenschaft (iv) folgt insbesondere $\|A^m\| \leq \|A\|^m$ für beliebiges $m \in \mathbb{N}$. Die Einheitsmatrix I besitzt stets die induzierte Matrixnorm $\|I\| = 1$.

Die für praktische Anwendungen wesentlichsten induzierten Matrixnormen ergeben sich aus der Maximumnorm und der Betragssummennorm:

Satz 1.3
(i) Die durch $\|x\|_\infty = \max\limits_{k=1(1)n} |x_k|$ *induzierte Matrixnorm ist*

$$\|A\|_\infty = \max_{i=1(1)n} \sum_{k=1}^{n} |a_{ik}| \qquad \textit{(maximale absolute Zeilensumme)}\,.$$

(ii) Die durch $\|x\|_1 = \sum\limits_{k=1}^{n} |x_k|$ *induzierte Matrixnorm ist*

$$\|A\|_1 = \max_{k=1(1)n} \sum_{i=1}^{n} |a_{ik}| \qquad \textit{(maximale absolute Spaltensumme)}\,.$$

BEWEIS: Für $A = O$ (Nullmatrix) gelten die Behauptungen. Sei nun $A \neq O$.
(i) Sei $x \in \mathbb{R}^n$ und $\alpha := \max\limits_{i=1(1)n} \sum_{k=1}^{n} |a_{ik}|$. Damit schätzt man ab

$$\begin{aligned}
\|Ax\|_\infty &= \max_{i=1(1)n} \left| \sum_{k=1}^{n} a_{ik} x_k \right| \\
&\leq \max_{i=1(1)n} \sum_{k=1}^{n} |a_{ik}|\, |x_k| \leq (\max_{i=1(1)n} \sum_{k=1}^{n} |a_{ik}|) \|x\|_\infty \\
&\leq \alpha \|x\|_\infty, \quad \text{also ist } \|A\|_\infty \leq \alpha.
\end{aligned}$$

Werde $\alpha = \max\limits_{i=1(1)n} \sum\limits_{k=1(1)n} |a_{ik}|$ für einen Index $\underline{i}$ erreicht, so definiert man x mit

$$x_k = \begin{cases} a_{\underline{i}k}/|a_{\underline{i}k}| & \text{für} \quad a_{\underline{i}k} \neq 0 \\ 0 & \text{für} \quad a_{\underline{i}k} = 0\,. \end{cases}$$

Offenbar ist $\|x\|_\infty = \max\limits_{k=1(1)n} |x_k| = 1$, und wir erhalten

$$\begin{aligned}
\alpha &= \max_{i=1(1)n} \sum_{k=1}^{n} |a_{ik}| = \sum_{k=1}^{n} |a_{\underline{i}k}| \\
&= \sum_{k=1}^{n} |a_{\underline{i}k} x_k| \quad \text{nach Definition von } x \\
&= |\sum_{k=1}^{n} a_{\underline{i}k} x_k| \quad \text{wegen } a_{\underline{i}k} \geq 0 \\
&\leq \max_{i=1(1)n} |\sum_{k=1}^{n} a_{ik} x_k| = \|Ax\|_\infty \\
&\leq \|A\|_\infty \|x\|_\infty \\
\alpha &\leq \|A\|_\infty,
\end{aligned}$$

also insgesamt $\|A\|_\infty = \alpha$.
(ii) Sei $x \in \mathbb{R}^n$ und $\beta := \max\limits_{k=1(1)n} \sum_{i=1}^{n} |a_{ik}|$. Abschätzung ergibt

$$\|Ax\|_1 = \sum_{i=1}^{n} |\sum_{k=1}^{n} a_{ik} x_k|$$

$$\begin{aligned} &\leq \sum_{k=1}^{n} \left(\sum_{i=1}^{n} |a_{ik}| \right) |x_k| \\ &\leq \sum_{k=1}^{n} \left(\max_{k=1(1)n} \sum_{i=1}^{n} |a_{ik}| \right) |x_k| \\ &\leq \beta \cdot \|x\|_1, \quad \text{also } \|A\|_1 \leq \beta. \end{aligned}$$

Werde $\beta = \max\limits_{k=1(1)n} \sum\limits_{i=1(1)n} |a_{ik}|$ für einen Index $\underline{k}$ erreicht, so erhält man unter Benutzung des Einheitsvektors $e_{\underline{k}}$:

$$\begin{aligned} \beta &= \max_{k=1(1)n} \sum_{i=1}^{n} |a_{ik}| \\ &= \sum_{i=1}^{n} |a_{i\underline{k}}| = \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{ij} \cdot \delta_{j\underline{k}}| \\ &= \|Ae_{\underline{k}}\|_1 \leq \|A\|_1 \|e_{\underline{k}}\|_1 = \|A_1\| \,, \end{aligned}$$

also $\beta \leq \|A\|_1$. Insgesamt ist damit $\|A\|_1 = \beta$. □

Die durch die Euklidische Vektornorm $\|x\|_2$ induzierte Matrixnorm ist die sogenannte *Spektralnorm*

$$\|A\|_2 = \sqrt{\max_{i=1(1)n} \mu_i} \,, \quad \mu_i \in \sigma(A^T A) \,.$$

Sie erfordert die Bestimmung von Eigenwerten μ_i der Produktmatrix $A^T A$ (des Gesamtspektrums $\sigma(A^T A)$ bzw. des betragsgrößten Eigenwertes) oder der Singulärwerte von A und ist deshalb für praktische Anwendungen wenig geeignet. Mitunter wird sie durch die dazu kompatible *Frobeniusnorm*

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik}^2}$$

wegen deren Eigenschaft $||A||_2 \leq ||A||_F$ ersetzt. Die Frobeniusnorm erfüllt zwar die 4 obigen Normeigenschaften, sie stellt jedoch keine induzierte Matrixnorm dar.

Um die untere Grenze aller Normen der Matrix $||A||$ zu ermitteln, benötigt man den Begriff des Spektralradius einer Matrix.

Definition 1.2 (Spektralradius) *Seien $\lambda_1, \lambda_2, \ldots, \lambda_n$ die Eigenwerte der (n,n)-Matrix A. Der* Spektralradius $\varrho(A)$ *ist das Maximum der Absolutbeträge aller Eigenwerte,also*

$$\varrho(A) := \max_{i=1(1)n} |\lambda_i| \,.$$

Der Spektralradius darf nicht mit der Spektralnorm $||A||_2$ verwechselt werden, denn nur für symmetrische Matrizen (bzw. hermitesche Matrizen im komplexen Fall) gilt allgemein $\varrho(A) = ||A||_2$. Für beliebige Matrizen stellt der Spektralradius eine untere Schranke aller Matrixnormen dar. Dies ergibt sich aus der Eigenwertgleichung $\lambda_i v_i = A v_i$ für jeden Eigenwert λ_i mit Eigenvektor $v_i \neq 0$, woraus

$$|\lambda_i| ||v_i|| = ||A v_i|| \leq ||A|| ||v_i||$$

folgt. Durch Maximumbildung über alle i erhält man die Schrankeneigenschaft von $\varrho(A)$.

Mit größerem Aufwand (vgl. [6]) läßt sich zudem nachweisen, daß der Spektralradius die untere Grenze aller induzierten Matrixnormen bildet.

Satz 1.4 *Bei gegebener Matrix $A \in \mathbb{R}^{n\times n}$ existiert zu jedem $\varepsilon > 0$ eine induzierte Matrixnorm $||A||$, so daß*

$$\varrho(A) \;\leq\; ||A|| \;\leq\; \varrho(A) + \varepsilon$$

gilt.

1.3 A-priori- und a-posteriori-Fehlerschätzungen

Wir betrachten wieder das Fixpunktproblem

$$x \;=\; g(x) \;\;,\quad g:\; D \subset \mathbb{R}^n \to \mathbb{R}^n$$

und setzen nun voraus, daß die Vektorfunktion

$$g(x) := \begin{pmatrix} g_1(x_1, x_2, \ldots, x_n) \\ g_2(x_1, x_2, \ldots, x_n) \\ \ldots\ldots\ldots\ldots \\ g_n(x_1, x_2, \ldots, x_n) \end{pmatrix}$$

stetig differenzierbar ist. Die Funktionalmatrix aller partiellen Ableitungen

$$G(x) = (g_{ij}(x)) = \left(\frac{\partial g_i(x)}{\partial x_j}\right) \tag{12}$$

heißt *Jacobi-Matrix* von g an der Stelle $x \in D \subset \mathbb{R}^n$. Unter der Voraussetzung stetiger Differenzierbarkeit der Abbildung g kann man leicht nachweisen, daß g in jedem Punkt $x \in D$ *Gateau-differenzierbar* ist und $G(x)$ die *Gateau-Ableitung* repräsentiert. Deshalb ist die Schreibweise $G(x) = g'(x)$ gerechtfertigt. Wegen der Stetigkeit der partiellen Ableitungen in x ist zudem garantiert, daß $g'(x)$ auch die *Fréchet-Ableitung* im Punkt x darstellt.

Mittels der Jacobimatrix kann man die Kontraktionsbedingung des Satzes 1.2 leicht durch folgende hinreichende Bedingung ersetzen:

Korollar 1.5

$$G(x) = g'(x) = \left(\frac{\partial g_i(x)}{\partial x_j}\right)$$

sei die Jacobimatrix von g an der Stelle $x \in \mathbb{R}^n$. Die Kontraktionsbedingung ist erfüllt, falls mit einer Matrixnorm

$$\|G(x)\| \;\leq\; \lambda \;<\; 1 \quad \textit{für alle} \quad x \in S \;\; \textit{gilt}\,.$$

Beweis: Für $x, y \in S$ gilt nach dem Mittelwertsatz

$$g(x) - g(y) = \int_0^1 G(tx + (1-t)y)\, dt\, (x - y)\,,$$

woraus durch Normabschätzung und Anwendung der Normeigenschaften der induzierten Matrixnorm auf der konvexen Menge S

$$\begin{aligned} ||g(x) - g(y)|| &\leq || \int_0^1 G(tx + (1-t)y)\, dt|| \cdot ||x - y|| && (13) \\ &\leq \int_0^1 ||G(tx + (1-t)y)||\, dt \cdot ||x - y|| && (14) \\ &\leq \lambda ||x - y|| && (15) \end{aligned}$$

folgt. □

Legt man die speziellen Matrixnormen $||.||_\infty$ oder $||.||_1$ zugrunde, so ergeben sich überprüfbare *Konvergenzkriterien*. Die Kontraktionsbedingung ist erfüllt, falls für alle $x \in S$ das Zeilensummen-Kriterium

$$\|G(x)\|_\infty \;=\; \max_{i=1(1)n} \;\sum_{j=1}^{n} |\frac{\partial g_i}{\partial x_j}| \;\leq\; \lambda \;<\; 1$$

oder das Spaltensummen-Kriterium

$$\|G(x)\|_1 \;=\; \max_{j=1(1)n} \;\sum_{i=1}^{n} |\frac{\partial g_i}{\partial x_j}| \;\leq\; \lambda \;<\; 1$$

nachgewiesen werden kann.

Beispiel 1.3 Das Nullstellenproblem aus Beispiel 1.2

$$\begin{aligned} f_1(x_1, x_2) &= 4x_1 - x_2 - x_1 \sin x_2 &&= 0 \\ f_2(x_1, x_2) &= (x_1 + x_2) \tan x_2 &&= 0 \end{aligned}$$

liefert nach Überführung in Fixpunktform $x = g(x)$ die Picard-Iteration

$$\begin{aligned} x_1^{k+1} &= \frac{1}{4}(x_1^k \sin x_2^k + x_2^k) \\ x_2^{k+1} &= \arctan \frac{4}{x_1^k + x_2^k}\,, \quad k = 0, 1, 2, \ldots . \end{aligned}$$

Die Jacobimatrix von g lautet damit

$$G(x) \;=\; \begin{pmatrix} \frac{\sin x_2}{4} & \frac{1 + x_1 \cos x_2}{4} \\ -\frac{4}{(x_1 + x_2)^2 + 16} & -\frac{4}{(x_1 + x_2)^2 + 16} \end{pmatrix}\,.$$

Mittels einer Skizze findet man die Startlösung $x^0 = (0.3, 1.0)^T$, die mit dem Zeilensummenkriterium die Norm

$$||G(x^0)||_\infty = \max(0.21037 + 0.29052, 0.22612 + 0.22612) = 0.50089 < 1$$

liefert. Wegen der Stetigkeit von $G(x)$ existiert dann eine Kugelumgebung von x^0 mit $||G(x)||_\infty \leq \lambda < 1$. Damit sind die Voraussetzungen des Satzes 1.2 zwar nicht vollständig verifiziert, aber eine Konvergenz der Iteration (10) ist zu erwarten. Dies bestätigen die errechneten Iterationswerte

k	x_1^k	x_2^k
0	0.3	1.0
1	0.313110324	1.256564428
2	0.388585753	1.196842600
3	0.389643331	1.193434330
4	0.388915579	1.193942318
5	0.388891626	1.193989819
6	0.388899633	1.193984729
7	0.388900040	1.193984099
8	0.388899954	1.193984147

Um eine a-posteriori-Schätzung des Fehlers der 8.Iterierten x^8 vorzunehmen, ist eine Näherung der Kontraktionskonstanten λ erforderlich. Einsetzen von x^8 in die Jacobimatrix ergibt $||G(x^8)||_\infty = 0.5183 < 0.52 =: \lambda$, womit man die Fehlerschranke

$$\begin{aligned} ||x^8 - x^*||_\infty &\leq \frac{\lambda}{1-\lambda}||x^8 - x^7||_\infty \\ &\leq \frac{0.52}{1-0.52} \cdot 8.56 \cdot 10^{-8} = 9.27 \cdot 10^{-8} \end{aligned}$$

erhält. Folglich ist der komponentenweise absolute Fehler kleiner als 10^{-7}, womit die Lösungswerte $x_1^* = 0.388900$, $x_2^* = 1.193984$ auf 6 Nachkommastellen genau geliefert werden. ◀

Während a-posteriori-Schätzungen in der Regel eine scharfe Fehlereinschließung liefern und als Genauigkeitsschranke der Näherungswerte praktikabel sind, überschätzen a-priori-Schranken den wahren Fehler oft beträchtlich. Mit ihrer Hilfe läßt sich jedoch zu Rechnungsbeginn (a-priori) die Anzahl k der Iterationsschritte abschätzen, die hinreichend sind, um eine vorgegebene Genauigkeit *tol* zu garantieren. Wird für den Näherungswert x_k die Fehlerschranke

$$\|x_k - x^*\| \;\leq\; tol \;=\; 10^{-m}$$

gefordert, so läßt sich dies mit der a-priori-Schätzung

$$\|x_k - x^*\| \leq \frac{\lambda^n}{1-\lambda}\|x_1 - x_0\| \leq 10^{-m}$$

erreichen, indem man sie nach k umstellt und so eine Schranke

$$k \geq \frac{\lg(\lambda) - lg\|x_1 - x_0\| - m}{\lg \lambda} \tag{16}$$

für die Iterationszahl erhält. Der einfachen Darstellung wegen soll dies nun an einem eindimensionalen Beispiel demonstriert werden.

Beispiel 1.4 Man bestimme die reellen Lösungen der Gleichung

$$x - \sin x \;=\; 0.25$$

mit einer Genauigkeit von 10^{-5} . Nach Überführung in die Fixpunktform $x = g(x) = \sin x + 0.25$ kann man mit einer Wertetafel für $g(x)$

x	0.9	1.0	1.1	1.2	1.3	1.5
g(x)	1.033	1.091	1.141	1.182	1.214	1.247

das Intervall $1.1 < x^* < 1.3$ für die gesuchte Lösung erkennen. Als Startnäherung wählen wir den Intervallmittelpunkt $x_0 = 1.2$. Die Voraussetzungen des Banachschen Fixpunktsatzes lassen sich hier leicht überprüfen. So ist $g(x)$ stetig differenzierbar mit $g'(x) = \cos x > 0$ für $0 < x < 1.5$ und damit streng monoton wachsend. Wegen $g(1.1) = 1.141$, $g(1.3) = 1.214$ folgt daraus $g : [1.1, 1.3] \to [1.1, 1.3]$. Eine geeignete Kontraktionskonstante erhält man wegen

$$\max_{x\in[a,b]} |g'(x)| = \max_{1.1\le x\le 1.3} |\cos x| = |\cos 1.1| = 0.4536$$

mit $\lambda = 0.4536 < 1$. Zur a-priori-Abschätzung der Iterationszahl k nutzt man die Werte $x_0 = 1.2$, $x_1 = g(x_0) = 1.182039086$, $\lambda = 0.4536$ und $m = 5$, die nach (16) eine Schranke

$$k \ge \frac{\lg(1 - 0.4536) - \lg(1.7961 \cdot 10^{-2}) - 5}{\lg 0.4536} \ge 10.24 \,,$$

ergeben, so daß $k = 11$ Iterationen hinreichend sind. Die Tabelle der Iterationswerte

k	x_k	$e_k = \lvert x_k - x^*\rvert$	e_{k+1}/e_k
0	1.2	2.877E-2	0.376
1	1.182039086	1.081E-2	0.385
2	1.175380828	4.151E-3	0.388
3	1.172836597	1.607E-3	0.388
4	1.171853595	6.239E-4	0.389
5	1.171472199	2.425E-4	0.389
6	1.171323981	9.433E-5	0.389
7	1.171266344	3.669E-5	0.389
8	1.171243926	1.427E-5	0.389
9	1.171235205	5.552E-6	0.389
10	1.171231812	2.159E-6	0.389
11	1.171230493	8.400E-7	0.389
..	.....	.....	.....
17	1.171229655		
18	1.171229654		
19	1.171229653		
20	1.171229653		

zeigt, daß der Fixpunkt $x^* = 1.171229653...$ im Rahmen der Rechengenauigkeit nach 20 Iterationen erreicht wird. Wegen $|x_{11} - x^*| \approx 8.4E-7$ sind 11 Iterationen hinreichend für die geforderte Genauigkeit. Jedoch sind nur 9 Iterationen notwendig!

Angenommen, es wurden 10 Iterationen ausgeführt. Dann liefert die a-priori-Schätzung eine Schranke

$$|x_{10} - x^*| \leq \frac{0.4536^{10}}{1 - 0.4536} |x_1 - x_0| \leq 1.212 \cdot 10^{-5} ,$$

wogegen die a-posteriori-Schätzung die schärfere Schranke

$$|x_{10} - x^*| \leq \frac{0.4536}{1 - 0.4536} |x_{10} - x_9| \leq 2.817 \cdot 10^{-6}$$

für den wahren Fehler $|x_{10} - x^*| = 2.159 \cdot 10^{-6}$ bestimmt. ◀

In der letzten Spalte der Tabelle werden die Quotienten zweier aufeinanderfolgender Fehler dargestellt. Wegen $e_{k+1}/e_k \to 0.389$ reduziert sich der Fehler asymptotisch, d.h. für $k \to \infty$, um diesen Faktor. Ein Iterationsverfahren mit konstanter positiver Konvergenzrate heißt *linear konvergent*. Die Frage, unter welchen Voraussetzungen auch im allgemeinen Fall

$$\lim_{k \to \infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||} = q \neq 0$$

gilt, wird im Abschnitt 2.3 behandelt.

Abschließend soll nun die Frage geklärt werden, ob die in Korollar 1.5 angegebene Normbedingung für die Jacobimatrix

$$\|G(x)\| \leq \lambda < 1 \quad \text{für alle} \quad x \in S$$

durch eine Punktbedingung an den Spektralradius $\varrho(G(x))$ ersetzt werden kann. Das gelingt nur bedingt, da nun die Existenz eines Fixpunktes vorausgesetzt werden muß.

Satz 1.6 (A.Ostrowski) *Angenommen, $g : D \subset \mathbb{R}^n \to \mathbb{R}^n$ habe einen Fixpunkt $x^* \in int(D)$ und sei stetig differenzierbar in D. Falls $\sigma = \varrho(G(x^*)) < 1$ ist, so existiert eine Kugel $S = S(x^*, \delta) = \{x \,|\, ||x - x^*|| \leq \delta\}$ um x^*, für die das Picard-Verfahren mit beliebigem $x^0 \in S$ gegen x^* konvergiert.*

Beweis: Zu $\varepsilon := (1 - \sigma)/2 > 0$ kann nach Satz 1.4 eine induzierte Matrixnorm mit $||G(x^*)|| \leq \sigma + \varepsilon$ gefunden werden. Darüber hinaus ist wegen der stetigen Differenzierbarkeit die Abbildung g Fréchet-differenzierbar in x^* . Damit existiert eine Kugel $S = S(x^*, \delta), \delta > 0$, so daß

$$||g(x) - g(x^*) - G(x^*)(x - x^*)|| \leq \varepsilon ||x - x^*|| \qquad \text{für alle} \quad x \in S$$

gilt. Dann folgt für alle $x \in S$

$$\begin{aligned} ||g(x) - g(x^*)|| &\leq ||g(x) - g(x^*) - G(x^*)(x - x^*)|| + ||G(x^*)||||x - x^*|| \\ &\leq (\varepsilon + \sigma)||x - x^*|| = \frac{1 + \sigma}{2} ||x - x^*|| . \end{aligned}$$

Anwendung auf $x^0 \in S$ ergibt wegen $\frac{1+\sigma}{2} < 1$

$$||x^k - x^*|| \leq \frac{1+\sigma}{2}||x^{k-1} - x^*|| \leq \cdots \leq \left(\frac{1+\sigma}{2}\right)^k ||x^0 - x^*|| < \delta$$

und $\lim_{k\to\infty} ||x^k - x^*|| = 0$. □

Auch dieser Satz liefert lediglich eine hinreichende Konvergenzbedingung, wogegen $\sigma < 1$ nicht notwendig ist.

Beispiel 1.5 (vgl.[11])
Im eindimensionalen Fall mit $g(x) = x - x^3$ ist $x^* = 0$ Fixpunkt. Obwohl $\sigma = g'(x^*) = 1$ gilt, konvergiert die Fixpunktiteration lokal gegen x^*.
Im Falle $g(x) = x + x^3$ gilt für $x^* = 0$ ebenfalls $\sigma = g'(x^*) = 1$. Hier divergiert jedoch die Fixpunktiteration bei beliebig nahe an x^* gewählten Startnäherungen. ◀

2 Fixpunktiteration für lineare Systeme

Alle bisherigen Betrachtungen treffen auch im Falle linearer Gleichungssysteme mit n Gleichungen und n Unbekannten

$$\begin{array}{ccccccc} a_{11}x_1 & + a_{12}x_2 + & \cdots & + a_{1n}x_n & = & a_1 \\ a_{21}x_1 & + a_{22}x_2 + & \cdots & + a_{2n}x_n & = & a_2 \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + a_{n2}x_2 + & \cdots & + a_{nn}x_n & = & a_n \end{array} \tag{17}$$

zu bzw. in Matrixnotation $Ax = a$ mit

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \; a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \; x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} .$$

Da sich die Konvergenzaussagen von Iterationsverfahren jedoch nun vereinfachen, sollen lineare Systeme gesondert behandelt werden. Die *direkten Verfahren* (Gauß-Algorithmus und seine Varianten, LU-Zerlegung, QR-Zerlegung, Cholesky-Zerlegung u.a.m.) lösen die Aufgabe (17) bei gut konditionierten Systemen kleiner und mittlerer Dimension n in endlich vielen Schritten zuverlässig, effektiv und numerisch stabil. Dagegen übersteigt der Rechenzeitbedarf für großdimensionale Gleichungssysteme wegen der Zeitkomplexität $O(n^3)$ der direkten Verfahren oft alle vertretbaren Grenzen. Eine Ausnahme bilden nur schwach besetzte Matrizen, z.B. die bei Diskretisierungen auftretenden Bandmatrizen und Block-Bandmatrizen.

Im allgemeinen Fall vollbesetzter Koeffizientenmatrizen sind deshalb *iterative Verfahren* die bessere Alternative, da der Aufwand pro Iterationsschritt mit $O(n^2)$ für eine Matrix-Vektor-Multiplikation beträchtlich geringer ist. Iterative Verfahren werden deshalb gegenwärtig in verschiendensten Varianten eingesetzt.

2.1 Allgemeine Konvergenzaussagen

Um die Resultate der vorhergehenden Abschnitte auf den Spezialfall linearer Systeme anwenden zu können, ist das Ausgangssystem (17) in Fixpunktgestalt (iterierfähige Gestalt) zu transformieren. Dies kann hier systematisch durchgeführt werden, indem man die Gleichungen nach ihren Hauptdiagonalelementen auflöst:

$$\begin{aligned}
x_1 &= -\frac{1}{a_{11}} \quad (-a_1 \qquad\quad +a_{12}x_2 \quad +\cdots \quad +a_{1n}x_n) \\
x_2 &= -\frac{1}{a_{22}} \quad (-a_2 \quad +a_{21}x_1 \qquad\quad +\cdots \quad +a_{1n}x_n) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n &= -\frac{1}{a_{nn}} \quad (-a_n \quad +a_{n1}x_1 \quad +a_{n2}x_2 \quad +\cdots \qquad\quad) \,.
\end{aligned} \tag{18}$$

Falls gewisse a_{ii} verschwinden, sind geeignete Zeilenvertauschungen vorzunehmen. Ausgehend von einer Startnäherung $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)^T$ erhält man durch Einsetzen in die rechte Seite eine neue Näherungslösung $x^1 = (x_1^1, x_2^1, \ldots, x_n^1)^T$ und bei Wiederholung die Iterationsvorschrift

$$x_i^{k+1} = \frac{1}{a_{ii}}(a_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^k), \quad i = 1(1)n \quad \text{für } k = 0, 1, 2, ...$$

Diese als *Jacobi-Verfahren* oder *Gesamtschrittverfahren* bekannte Fixpunktiteration kann man leicht verallgemeinern. Wir setzen dazu voraus, daß $detA \neq 0$ ist und damit das lineare Gleichungssystem

$$Ax \;=\; a \tag{19}$$

eine eindeutige Lösung x^* besitzt. Dann läßt sich die Koeffizientenmatrix auf beliebig viele Arten in der Form

$$A = N - P \qquad \text{mit} \quad detN \neq 0$$

aufspalten, wobei N und P ebenfalls (n, n)-Matrizen sind. Das System kann nun in

$$Nx = Px + a$$

umgeformt werden bzw. in

$$x = N^{-1}Px + N^{-1}a \,.$$

Mit den Bezeichnungen $B = N^{-1}P$ und $b = N^{-1}a$ erhält man daraus das System in *iterierfähiger Gestalt*

$$x = Bx + b \,. \tag{20}$$

Beginnend mit einem Startvektor x^0, definiert man eine Folge $\{x^k\}$ von Vektoren durch

$$\boxed{x^{k+1} = Bx^k + b \,, \quad k = 1, 2, 3, ...} \tag{21}$$

Im Vergleich zur allgemeinen Picard-Iteration treten in der affinen Abbildung g mit

$$g(x) = Bx + b$$

die Komponenten von x nur linear auf, so daß hier ein *lineares Einschrittverfahren* vorliegt.

Definition 2.1 (Iterationsverfahren)

(i) Die Matrix N^{-1} heißt erzeugende Matrix *des Iterationsverfahrens.*

(ii) Die Matrix $B = N^{-1}P$ heißt Iterationsmatrix *des Iterationsverfahrens.*

(iii) Das Iterationsverfahren (21) ist stationär*, denn B und b hängen nicht vom Index k ab.*

Wegen der Darstellung $g(x) = Bx + b$ ist die Abbildung g stetig differenzierbar mit der konstanten Jacobimatrix $G(x) = B$. Damit vereinfachen sich die Aussagen des Satzes 1.2 und des Korollars 1.5 und liefern folgenden

Satz 2.1

Mit einer beliebigen Matrixnorm sei $||B|| < 1$ erfüllt. Dann gelten folgende Behauptungen:

(i) Gleichung (20) besitzt eine eindeutige Lösung (den einzigen Fixpunkt von g) x^* .

(ii) Die Iterationsvektoren x^k des Verfahrens (21) konvergieren bei beliebig gewähltem Startvektor x^0 stets gegen die Lösung x^* .

(iii) Für den k-ten Iterationsvektor x^k gilt mit $\lambda = ||B||$ die a-priori–Fehlerschätzung

$$\|x^k - x^*\| \leq \frac{\lambda^n}{1-\lambda}\|x^1 - x^0\|$$

sowie die a-posteriori–Fehlerschätzung

$$\|x^k - x^*\| \leq \frac{\lambda}{1-\lambda}\|x^k - x^{k-1}\| \; .$$

Beweis: Mit der Kontraktionskonstanten $\lambda := ||G(x)|| = ||B|| < 1$ ist Korollar 1.5 global auf ganz $\mathbb{R}^n$ erfüllt. An die Startlösung x^0 muß keine einschränkende Bedingung gestellt werden, so daß Satz 1.2 nun in $\mathbb{R}^n$ gültig ist, womit auch die Fehlerschätzungen folgen. □

Ein zugehöriger Algorithmus `linear` für das lineare Einschrittverfahren (21) setzt als Input die Iterationsmatrix B, den Vektor b, einen Startwert x und die (absolute und relative) Toleranz $tolabs, tolrel$ voraus.

Algorithmus 2.1 (Lineares Einschritt-Verfahren)

Function linear$(B, b, x, tolabs, tolrel)$

1. Berechne eine Norm $\lambda = ||B||$. Falls $\lambda > 1$, so Stop.
2. Berechne $y = Bx + b$ und die Toleranz $tol = tolrel \cdot ||y|| + tolabs$
3. Do while $\frac{\lambda}{1-\lambda}||y - x|| > tol$
   1. Überspeichere $x = y$
   2. Berechne $y = Bx + b$
   3. Aktualisiere $tol = tolrel \cdot ||y|| + tolabs$
4. Return y

Für die Konvergenz sind offenbar allein die Eigenschaften der Iterationsmatrix B, nicht aber die des Vektors b ausschlaggebend. Es genügt, wenn für irgendeine Matrixnorm der Nachweis $\|B\| < 1$ erfolgt. Das kann auch für eine kompatible Norm, z. B. die Frobeniusnorm $\|B\|_F$ gelten, denn wegen $\|B\|_2 \leq \|B\|_F < 1$ ist dann die Voraussetzung mit der Spektralnorm $\|B\|_2$ erfüllt und die Konvergenz in der Euklidischen Norm garantiert.

Mit den speziellen Matrixnormen $||.||_\infty$ oder $||.||_1$ lautet nun das Zeilensummen-Kriterium

$$\|B\|_\infty \;=\; \max_{i=1(1)n} \;\sum_{j=1}^{n} |b_{ij}| \;<\; 1$$

und das Spaltensummen-Kriterium

$$\|B\|_1 \;=\; \max_{j=1(1)n} \;\sum_{i=1}^{n} |b_{ij}| \;<\; 1\,.$$

Wie bereits im allgemeinen (nichtlinearen) Fall erweisen sich diese Normbedingungen als hinreichend, nicht jedoch als notwendig für die Konvergenz gegen eine Lösung. Ein notwendiges und hinreichendes Konvergenzkriterium erhält man wiederum mit Hilfe des Spektralradius der Iterationsmatrix B (vgl. [11]).

Satz 2.2 *Die affine Abbildung g mit* $g(x) = Bx + b$ *besitzt einen eindeutigen Fixpunkt* x^* *in* $\mathbb{R}^n$*, und die Iterationsvektoren* x^k *der Fixpunktiteration (21) konvergieren bei beliebig gewähltem Startvektor* x^0 *stets gegen die Lösung* x^* *genau dann, wenn* $\varrho(B) < 1$ *ist.*

2.2 Iteration in Gesamtschritten (Jacobi-Verfahren)

Die nachfolgenden Iterationsverfahren sind Spezialfälle des allgemeinen Verfahrens

$$x^k = Bx^{k-1} + b, \quad B = N^{-1}P, \quad b = N^{-1}a$$

und unterscheiden sich lediglich in der Wahl von N und P. Das eingangs beschriebene Verfahren einer Auflösung nach den Hauptdiagonalelementen von A ergibt

$$\begin{array}{llllllll}
a_{11}x_1 & & = & & -a_{12}x_2 & -\cdots- & a_{1n}x_n & +a_1 \\
& a_{22}x_2 & = & -a_{21}x_1 & & -\cdots- & a_{2n}x_n & +a_2 \\
\multicolumn{8}{c}{\dotfill} \\
& a_{nn}x_n & = & -a_{n1}x_1 & & -\cdots & & +a_n
\end{array}$$

bzw.

$$\underbrace{\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \multicolumn{4}{c}{\dotfill} \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}}_{N} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & 0 & \cdots & -a_{2n} \\ \multicolumn{4}{c}{\dotfill} \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{pmatrix}}_{P} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{pmatrix} + \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ a_n \end{pmatrix}}_{a}$$

A wird also in die Diagonalmatrix N und die Restmatrix P zerlegt. Das *Gesamtschrittverfahren (Jacobi-Verfahren, J-Verfahren)* lautet dann

$$\boxed{x_i^k = \frac{1}{a_{ii}} \left(a_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{k-1} \right), \quad i = 1(1)n\,.} \tag{22}$$

Da die Iterationsmatrix $B = N^{-1}P$ die Elemente

$$b_{ij} = \begin{cases} -a_{ij}/a_{ii} & \text{für} \quad i \neq j \\ 0 & \text{für} \quad i = j \end{cases}$$

besitzt, erhalten die eingeführten *Konvergenzkriterien* die einfache Form

$$\sum_{\substack{j=1 \\ j \neq j}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1 \qquad \text{für alle } i = 1(1)n \tag{23}$$

für das Zeilensummenkriterium bzw.

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1 \qquad \text{für alle } j = 1(1)n \tag{24}$$

für das Spaltensummenkriterium.

Beispiel 2.1 Für das lineare System

$$\begin{aligned} 12x_1 + 2x_2 + 3x_3 &= 18 \\ -x_1 + 8x_2 + 2x_3 &= -32 \\ x_1 + 3x_2 + 12x_3 &= 6 \end{aligned}$$

lautet das Gesamtschrittverfahren

$$\begin{aligned} x_1^{k+1} &= 1.5 - 0.1667\, x_2^k - 0.25\, x_3^k \\ x_2^{k+1} &= -4.0 + 0.125\, x_1^k - 0.25\, x_3^k \\ x_3^{k+1} &= 0.5 - 0.0833\, x_1^k + 0.25\, x_2^k\,. \end{aligned}$$

Bezeichnet man $b_i = a_i/a_{ii}$, $b_{ij} = -a_{ij}/a_{ii}$, so erhält man für die Koeffizienten b_{ij} das Schema auf der folgenden Seite.
Näherungslösungen sind damit $x_1 = 2.2410$, $x_2 = -3.5748$, $x_3 = -0.5804$. Für die Iterationsmatrix

$$B = \begin{pmatrix} 0 & -0.1667 & -0.25 \\ 0.125 & 0 & -0.25 \\ -0.0833 & 0.25 & 0 \end{pmatrix}$$

ist wegen

$$\begin{aligned} \|B\|_\infty &= \max_i \sum_j |b_{ij}| = 0.4167 < 1 \qquad \text{und} \\ \|B\|_1 &= \max_j \sum_i |b_{ij}| = 0.5 < 1 \end{aligned}$$

die Konvergenz bei beliebiger Startnäherung garantiert.

b_i	b_{i1}	b_{i2}	b_{i3}
1.5	*	-0.1667	-0.2500
-4.0	0.125	*	-0.2500
0.5	-0.0833	0.2500	*
k	x_1^k	x_2^k	x_3^k
0	0	0	0
1	1.5000	-4.0000	0.5000
2	2.0418	-3.9375	-0.6250
3	2.3126	-3.5885	-0.6545
4	2.2618	-3.5473	-0.5898
5	2.2388	-3.5698	-0.5752
6	2.2389	-3.5764	-0.5789
7	2.2409	-3.5754	-0.5806
8	2.2411	-3.5747	-0.5805
9	2.2410	-3.5747	-0.5804
10	2.2410	-3.5748	-0.5804

In der Maximumnormorm $\|x\|_\infty$ ergibt sich mit den Konstanten $\lambda = \|B\|_\infty = 0.4167$, $\|x^1 - x^0\| = 4.0$ die a-priori-Abschätzung

$$\|x^k - x^*\|_\infty \;\leq\; \frac{0.4167^k}{1 - 0.4167} \cdot 4 \;=:\; \alpha_k.$$

Durch Vergleich dieser Schranken α_k mit den tatsächlichen Fehlergrößen

k	α_k	$\|x^k - x^*\|_\infty$
2	1.191	0.3627
4	0.2068	0.0275
6	0.0359	0.0021
8	0.0062	0.0001

erkennt man die starke Überschätzung des Fehlers durch die a-priori-Abschätzung. ◄

Das Gesamtschrittverfahren kann in Matrixschreibweise angegeben werden, wenn man die additive Aufspaltung von A mittels $A = D + L + R$ in die Matrizen

$$D = \mathrm{diag}(a_{11}, a_{22}, ..., a_{nn})\,,$$

$$L = \begin{pmatrix} 0 & \cdots & & \\ a_{21} & 0 & & \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

vornimmt. Mit $N = D$ und $P = -(L + R)$ lautet das Verfahren dann

$$\boxed{x^{k+1} = D^{-1}[a - (L + R)x^k], \quad k = 0, 1, 2, \ldots.} \tag{25}$$

2.3 Iteration in Einzelschritten (Gauß-Seidel-Verfahren)

Wendet man zur Bestimmung des Iterationswertes x_i^k jeweils die zuletzt bestimmten Werte der anderen Komponenten x_j an, so ergibt sich folgende Verfahrensvariante

$$\begin{aligned}
x_1^k &= \frac{1}{a_{11}}(a_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \ldots - a_{1n}x_n^{k-1}) \\
x_2^k &= \frac{1}{a_{22}}(a_2 - a_{21}x_1^k - a_{23}x_3^{k-1} - \ldots - a_{1n}x_n^{k-1}) \\
x_3^k &= \frac{1}{a_{33}}(a_3 - a_{31}x_1^k - a_{32}x_2^k - \ldots - a_{1n}x_n^{k-1}) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^k &= \frac{1}{a_{nn}}(a_n - a_{n1}x_1^k - a_{n2}x_2^k - \ldots - a_{n,n-1}x_{n-1}^k)\,.
\end{aligned} \tag{26}$$

In Kurzform lautet dieses *Gauß-Seidel-Verfahren*

$$x_i^k = \frac{1}{a_{ii}}(a_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^{n} a_{ij}x_j^{k-1})\,,\ i = 1(1)n \quad \text{für} \quad k = 1, 2, 3, \ldots. \tag{27}$$

Da man nach jeder Bestimmung einer Komponente x_i^k des Näherungsvektors x^k den im Speicher stehenden Vektor x_i aufdatiert, wird x^k in jedem Einzelschritt verändert, woher auch der Name *Einzelschrittverfahren* kommt. Dem Gesamtschritt entspricht dann die Abarbeitung eines gesamten Iterationszyklus für $i = 1(1)n$.

Aus (27) folgt übrigens

$$a_{ii}x_i^k + \sum_{j=1}^{i-1} a_{ij}x_j^k + \sum_{j=i+1}^{n} a_{ij}x_j^{k-1} = a_i$$

und damit die Darstellung in Matrizenform

$$\begin{pmatrix} a_{11} & & & \\ & a_{22} & & 0 \\ & & \ddots & \\ 0 & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ \cdot \\ x_n^k \end{pmatrix} + \begin{pmatrix} 0 & 0 & \cdots & 0 \\ +a_{21} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ +a_{n1} & +a_{n2} & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ \cdot \\ x_n^k \end{pmatrix}$$

$$- \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \cdots & -a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1^{k-1} \\ x_2^{k-1} \\ \cdot \\ x_n^{k-1} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ a_n \end{pmatrix}.$$

Die Matrix A wird also in folgende Matrizen zerlegt:

$$A = \underbrace{\begin{pmatrix} a_{11} & & & & \\ a_{21} & a_{22} & & 0 & \\ a_{31} & a_{32} & a_{33} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}}_{N} - \underbrace{\begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ 0 & 0 & -a_{23} & \cdots & -a_{2n} \\ 0 & 0 & 0 & \cdots & -a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}}_{P}$$

$$A = \qquad\qquad N \qquad\qquad - \qquad\qquad P$$

Offenbar ist $\det N \neq 0$, falls $a_{ii} \neq 0$, $i = 1(1)n$ ist. Daß die Konvergenzkriterien (23) und (24) des Gesamtschrittverfahrens auch hinreichend für die Konvergenz des Gauß-Seidel-Verfahrens sind, zeigt folgender

Satz 2.3 *Zeilensummenkriterium (23) und Spaltensummenkriterium (24) sind auch hinreichend für die Konvergenz des Gauß-Seidel-Verfahrens.*

Beweis: Wir beweisen die Behauptung für das Zeilensummenkriterium. Mit $b_{ij} := a_{ij}/a_{ii}$, $b_i := a_i/a_{ii}$ lautet das Verfahren

$$x_i^k = b_i - \sum_{j=1}^{i-1} b_{ij} x_j^k - \sum_{j=i+1}^{n} b_{ij} x_j^{k-1} .$$

Die Lösung x^* erfüllt die Fixpunktgleichung

$$x_i^* = b_i - \sum_{j=1}^{i-1} b_{ij} x_j^* - \sum_{j=i+1}^{n} b_{ij} x_j^* .$$

Subtraktion beider Beziehungen liefert für die Fehlergrößen $e_i^k := x_i^k - x_i^*$ die Abschätzungen

$$\begin{aligned} e_i^k &= -\sum_{j=1}^{i-1} b_{ij} e_j^k - \sum_{j=i+1}^{n} b_{ij} e_j^{k-1} \\ |e_i^k| &\leq \sum_{j=1}^{i-1} |b_{ij}||e_j^k| + \sum_{j=i+1}^{n} |b_{ij}||e_j^{k-1}| \\ &\leq \sum_{j=1}^{i-1} |b_{ij}| \cdot \|e^k\|_\infty + \sum_{j=i+1}^{n} |b_{ij}| \cdot \|e^{k-1}\|_\infty \\ |e_i^k| &\leq p_i \cdot \|e^k\|_\infty + q_i \cdot \|e^{k-1}\|_\infty, \quad i = 1(1)n \end{aligned}$$

mit $p_i = \sum_{j=1}^{i-1} |b_{ij}|, \quad q_i = \sum_{j=i+1}^{n} |b_{ij}|$.

Sei $s = s(k)$ derjenige Indexwert für i, für den $|e_s^k| = max|e_i^k| = \|e^k\|_\infty$ angenommen wird. Dann gilt offenbar für $i = s$ wegen Voraussetzung $p_s < 1$

$$\|e^k\|_\infty \leq p_s \cdot \|e^k\|_\infty + q_s \|e^{k-1}\|_\infty \qquad \text{bzw.} \qquad \|e^k\|_\infty \leq \frac{q_s}{1 - p_s} \|e^{k-1}\|_\infty .$$

Es bleibt nun noch zu zeigen, daß mit der Iterationsmatrix B des *Gesamtschrittverfahrens*

$$\lambda := \max_s \frac{q_s}{1-ps} \leq \|B\|_\infty = \max_i \sum_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}| < 1$$

gilt. Da aber nach Voraussetzung

$$p_i + q_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}| \leq \|B\|_\infty < 1$$

ist, folgt $q_i \leq \|B\|_\infty - p_i$ und hieraus

$$\frac{q_i}{1-p_i} \leq \frac{\|B\|_\infty - p_i}{1-p_i} \leq \frac{\|B\|_\infty - p_i\|B\|_\infty}{1-p_i} = \|B\|_\infty.$$

Deshalb ist mit $\lambda := \max_i \frac{q_i}{1-p_i}$ die Ungleichung

$$\lambda \leq \|B\|_\infty < 1$$

erfüllt. Der Beweis für das Spaltensummenkriterium verläuft ähnlich. □

Folgerung 2.4 *Wegen* $\lambda := \max_s \frac{q_s}{1-p_s} \leq \|B\|_\infty$

konvergiert das Einzelschrittverfahren bei Erfülltsein des Zeilensummenkriteriums im allgemeinen schneller als das Gesamtschrittverfahren.

Beispiel 2.2 Wir betrachten Beispiel 2.1. Das Einzelschrittverfahren lautet

$$\begin{array}{rcrcrcrcl} x_1^{k+1} = & & 1.5 & & & - & 0.1667\, x_2^k & - & 0.2500\, x_3^k \\ x_2^{k+1} = & -4.0 & + & 0.125\, x_1^{k+1} & & & & - & 0.2500\, x_3^k \\ x_3^{k+1} = & & 0.5 & - & 0.0833\, x_1^{k+1} & + & 0.2500\, x_2^{k+1} & & \end{array} \tag{28}$$

Mit dem Schema

b_i	b_{i1}	b_{i2}	b_{i3}
1.5	*	-0.1667	-0.2500
-4.0	0.125	*	-0.2500
0.5	-0.0833	0.2500	*
k	x_1^k	x_2^k	x_3^k
0	0	0	0
1	1.5	-5.8125	0.5781
2	2.27996	-3.5705	-0.5826
3	2.2407	-3.5743	-0.5803
4	2.2408	-3.5748	-0.5804
5	2.2409	-3.5748	-0.5804

erhält man nach 5 Iterationen die Näherungen

$$x_1 = 2.2409, \quad x_2 = -3.5748, \quad x_3 = -0.5804\,.$$

Das Zeilensummenkriterium liefert $\|B\|_\infty = 0.4167$. Am Schema erkennt man, daß der Konvergenzfaktor $\lambda = \max\limits_s \frac{q_s}{1-p_s} = 0.4167$ nicht kleiner als beim Gesamtschrittverfahren ist. Dennoch konvergiert das GS-Verfahren mit 5 Schritten wesentlich rascher als das J-Verfahren, das 10 Schritte benötigt. ◀

Zur Gewinnung der Matrixschreibweise des Einzelschrittverfahrens zerlegt man die Koeffizientenmatrix additiv $A = D + L + R$ mit den bereits eingeführten Matrizen und erhält

$$\begin{aligned} Dx + Lx + Rx &= a \\ (D+L)x &= a - Rx \\ x &= (D+L)^{-1}[a - Rx]\,, \end{aligned}$$

womit $N = D + L$ und $P = -R$ gilt und damit die Iterationsmatrix $B = N^{-1}P = -(D+L)^{-1}R$ lautet. Aufgelöst nach x^{k+1} ergibt sich die erste Form des GS-Verfahrens

$$\boxed{x^{k+1} = (D+L)^{-1}[a - Rx^k], \quad k = 0, 1, 2, \ldots.} \tag{29}$$

Wegen der aufwendigen Matrixinversion $(D+L)^{-1}$ ist diese Darstellung keinesfalls für praktische Rechnungen zu empfehlen; hier sollte die zweite Form (27) in Matrixform

$$x^{k+1} = D^{-1}[a - Lx^{k+1} - Rx^k], \quad k = 0, 1, 2, \ldots$$

benutzt werden.

Die Fixpunktiteration für lineare Gleichungssysteme erfordert eine solche iterierfähige Form, die ein konvergentes Verfahren liefert. Für zahlreiche Anwendungsprobleme ist eine geeignete Umstellung tatsächlich leicht möglich, insbesondere für strikt diagonal-dominante Matrizen.

Definition 2.2 (Strikte Diagonaldominanz) $A \in \mathbb{R}^{n\times n}$ *ist* strikt diagonal-dominant*, falls gilt:*

$$\sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| \;<\; |a_{ii}|\,, \quad i = 1(1)n\,.$$

Offenbar sind die Diagonalelemente a_{ii} derartiger Koeffizientenmatrizen stets ungleich Null, so daß Gesamt- und Einzelschrittverfahren ohne Zeilenvertauschung anwendbar sind. Darüberhinaus ist offensichtlich das Zeilensummenkriterium

$$\sum_{\substack{j=1\\ j\neq j}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| < 1 \qquad \text{für alle } i = 1(1)n$$

erfüllt, womit nach Satz 2.1 und Satz 2.3 beide Iterationsverfahren gegen die eindeutige Lösung x^* konvergieren. Damit hat man folgenden Satz bewiesen:

Satz 2.5 *Ist die Koeffizientenmatrix A strikt diagonal-dominant, so ist A regulär und das lineare Gleichungssystem eindeutig lösbar. Gesamt- und Einzelschrittverfahren sind konvergent.*

Das konstruierte System in Beispiel 2.1 ist offenbar strikt diagonal-dominant. Jedoch tritt diese Eigenschaft häufig gerade bei Problemklassen linearer Differentialgleichungen auf.

Beispiel 2.3 Wir betrachten lineare Randwertprobleme 2.Ordnung der Form

$$-\frac{d^2x}{dt^2} + p(t)x = q(t)\,, \quad x(a) = x_a,\ x(b) = x_b$$

auf dem Intervall $I = [a, b]$. Die Funktionen p, q seien stetig, und es existiere eine Konstante $Q > 0$, so daß $p(t) \geq Q > 0$ auf I gilt. Dann besitzt das Problem nach [6] eine eindeutige Lösung $x \in C^2(I)$.

Man kann diese Lösung $x(t)$ auf den Gitterpunkten $t_i = a + ih$, $i = 0(1)n$, mit der Schrittweite $h = (b - a)/n$ durch Näherungen x_i approximieren, indem man das gegebene Problem durch die endlichdimensionale Aufgabe am Gitterpunkt t_i

$$-\frac{1}{h^2}(x_{i-1} - 2x_i + x_{i-1}) + p(t_i)x_i \;=\; q(t_i)$$

ersetzt. Mit den Abkürzungen $a_i := 2 + h^2p(t_i)$ und $b_i := h^2q(t_i)$ ergibt sich nach Zusammenfassung ein lineares Gleichungssystem

$$-x_{i-1} + a_ix_i - x_{i+1} \;=\; b_i\,, \quad i = 1(1)n - 1\,. \tag{30}$$

Die beiden Lösungskomponenten $x_0 = x_a$ und $x_n = x_b$ sind durch die Randbedingungen x_a und x_b vorgegeben und können in die erste und letzte Gleichung eingesetzt werden. Um eine genaue Approximation der Lösung zu erhalten, ist n hinreichend groß zu wählen (z.B. $n = 500$). Die Koeffizientenmatrix A hat tridiagonale Form und ist wegen der Abschätzung $|a_i| = 2 + h^2p(t_i) \geq 2 + h^2Q > 2$ strikt diagonal-dominant. Damit ist System (30) bei beliebiger fester Dimension n eindeutig lösbar und beide Iterationsverfahren konvergieren gegen diese Lösung. Allerdings wird die Konvergenz wegen

$$||B||_\infty = \max_i \left|\frac{2}{a_i}\right| = \max_i \frac{2}{2 + h^2p(t_i)} \;\to\; 1 \quad \text{für} \quad h \to 0$$

bei zunehmender Approximationsgenauigkeit immer langsamer verlaufen. ◀

In [14] werden *Relaxationsverfahren* als Verallgemeinerungen der beiden Verfahrensklassen behandelt. Mit sukzessiver Überrelaxation auf Basis des Jacobi-Verfahrens (JOR) bzw. des Gauß-Seidel-Verfahrens (SOR) lassen sich Konvergenzbeschleunigungen bei speziell strukturierten Gleichungssystemen erreichen. Eine sehr detaillierte und umfangreiche Darstellung der Iterationsverfahren für *lineare Systeme* findet man in [5].

3 Linearisierung und Newton-Verfahren

3.1 Konvergenzordnung und überlinear konvergente Verfahren

Die Fixpunktiteration ist im allgemeinen ein linear konvergentes Verfahren. Dies konnte bei der Analyse des Beispiels 1.4 festgestellt werden, die durch ein asymptotisches Verhalten des Fehlers

$$\lim_{k\to\infty} ||x^{k+1} - x^*||/||x^k - x^*|| = q$$

mit einer positiven Konstanten q charakterisiert war. Um überlineares Konvergenzverhalten auch quantitativ zu beschreiben, wurde der Begriff der Konvergenzordnung p eingeführt (vgl. insbesondere [10] und [13]). Wir betrachten dazu eine Folge von Näherungen $\{x^k\}$ für den Fixpunkt x^* , die mit der Iteration $x^{k+1} = g(x^k)$ erzeugt wird.

Definition 3.1 (Q-Ordnung)

(i) *Die Folge* $\{x^k\}$ *konvergiert* mindestens mit der Q-Ordnung $p \geq 1$, *falls ein Konvergenzfaktor* $Q_+ > 0$ *existiert, so daß für alle* $k \geq k_0$, $k_0 \in \mathbb{N}$,

$$\|x^{k+1} - x^*\| \;\leq\; Q_+\|x^k - x^*\|^p$$

gilt. Falls $p = 1$ *ist, so wird* $Q_+ < 1$ *gefordert.*

(ii) *Die Folge* $\{x^k\}$ *konvergiert* höchstens mit der Q-Ordnung $p \geq 1$, *falls ein Konvergenzfaktor* $Q_- > 0$ *existiert, so daß für alle* $k \geq k_0$, $k_0 \in \mathbb{N}$,

$$\|x^{k+1} - x^*\| \;\geq\; Q_-\|x^k - x^*\|^p$$

gilt.

(iii) *Die Folge* $\{x^k\}$ *konvergiert* genau mit der Q-Ordnung $p \geq 1$, *falls sie mindestens und höchstens mit Q-Ordnung p konvergiert.*

(iv) *Das Iterationsverfahren* $x^{k+1} = g(x^k)$ *konvergiert bezüglich* x^* mindestens mit der Q-Ordnung p , *wenn jede damit erzeugte Folge* $\{x^k\}$ *mindestens mit Q-Ordnung p konvergiert.*

(v) *Existiert eine Folge* $\{x^k\}$ *genau mit Q-Ordnung p, so konvergiert auch das Iterationsverfahren* $x^{k+1} = g(x^k)$ *bezüglich* x^* genau mit Q-Ordnung p .

Verfahren der Q-Ordnungen 1, 2 bzw. 3 werden als *linear, quadratisch bzw. kubisch konvergent* bezeichnet. Das trifft für die beiden Fälle „mindestens“ und „genau“ des Teiles (iv) zu.

Definition 3.1 setzt allerdings nicht voraus, daß die Fehler $\|x^k - x^*\|$ verschieden von Null sind. Tritt dieser Fall für ein $k = k_1 > k_0$ bei einem Verfahren ein, das mit *genau* der Q-Ordnung p konvergiert, so verschwinden alle Fehler der Iterationen für $k \geq k_1$ (trivialer Fall). Andererseits bedeutet $\|x^k - x^*\| > 0$ für ein $k \geq k_0$, daß alle weiteren Fehler verschieden von Null sind .

Wir setzen desweiteren diesen nichttrivialen Fall und ein Verfahren der genauen Q-Ordnung p voraus. Existiert der Grenzwert

$$q := \lim_{k\to\infty} \frac{||x^{k+1} - x^*||}{||x^k - x^*||^p} ,$$

so bezeichnet man ihn als *asymptotischen Konvergenzfaktor*. Falls jedoch $q = 0$ ist, so erweist sich das Verfahren als *schneller als mit Q-Ordnung p konvergent*. Insbesondere heißen Iterationsverfahren mit

$$\lim_{k\to\infty} \frac{||x^{k+1} - x^*||}{||x^k - x^*||} = 0$$

überlinear konvergent.

Das weitere Ziel besteht nun darin, spezielle Iterationsverfahren zu entwickeln, die überlineare – also mindestens quadratische – Konvergenz besitzen. Dazu ist allerdings Definition 3.1 wenig praktikabel. Für zweimal stetig differenzierbare Funktionen g kann folgendes einfachere Kriterium aufgestellt werden.

Satz 3.1 *Ist g zweimal stetig differenzierbar und gilt für die Jacobimatrix $G(x^*) = O$ (Nullmatrix), so konvergiert die Picard-Iteration $x^{k+1} = g(x^k)\,,\ k = 0, 1, 2, \ldots\,,$ mindestens Q-quadratisch.*

Beweis: Sei $S = \{x \,|\, ||x - x^*|| \le \varrho\}$ eine abgeschlossene Kugel um x^*, in der alle Iterierten x^k für $k \ge k_0$ liegen. Für den Fehler der $(k+1)-$ten Iterierten erhält man mit dem Taylorschen Satz für Funktionen $g : \mathbb{R}^n \to \mathbb{R}^n$ die Darstellung

$$\begin{aligned} ||x^{k+1} - x^*|| &= ||g(x^k) - g(x^*)|| \\ &= ||g(x^*) + g'(x^*)(x^k - x^*) + R(x^k, x^*) - g(x^*)|| \\ &= ||R(x^k, x^*)|| \end{aligned}$$

wegen der Voraussetzung $g'(x^*) = G(x^*) = O$. Mit der Darstellung des Restgliedes $R(x^k, x^*)$ liefern die Normabschätzungen

$$\begin{aligned} ||x^{k+1} - x^*|| &= \left\| \int_0^1 (1-t) g''(tx^k + (1-t)x^*)(x^k - x^*)^2 \, dt \right\| \\ &\le \int_0^1 (1-t)\, ||g''(tx^k + (1-t)x^*)||\, dt \ \cdot ||x^k - x^*||^2 \\ &\le \max_{x\in S} ||g''(x)|| \cdot \int_0^1 (1-t)\, dt \ \cdot ||x^k - x^*||^2 \\ &\le Q||x^k - x^*||^2 \end{aligned}$$

mit einer Konstanten $Q \ge 0$. □

Den beträchtlichen Vorteil einer Konvergenzordnung $p > 1$ verdeutlicht man am besten, wenn man die Definition

$$\|x^{k+1} - x^*\| \ \le\ Q\|x^k - x^*\|^p$$

rekursiv anwendet, was zur Darstellung

$$\|x^k - x^*\| \ \le\ Q^{\frac{p^k-1}{p-1}} \|x^0 - x^*\|^{p^k} \tag{31}$$

führt. Bei linearer Konvergenz hätte sich dagegen

$$\|x^k - x^*\| \ \le\ Q^k \cdot \|x^0 - x^*\| \tag{32}$$

ergeben. Angenommen, in beiden Fällen liege derselbe Anfangsfehler $\|x^0 - x^*\| = 1.0$ vor und die Konvergenzkonstante sei $Q = 0.1$. Dann schätzt man den Fehler des linear konvergenten Verfahrens durch

$$\|x^k - x^*\| \leq 10^{-k}$$

ab, d.h. pro Iterationsschritt gewinnt man (asymptotisch) 1 Dezimalziffer. Für ein quadratisch konvergentes Verfahren dagegen erhält man

$$\|x^k - x^*\| \leq 10^{-2^k} ,$$

so daß sich in jedem Iterationsschritt die Zahl der richtigen Ziffern verdoppelt!

Beispiel 3.1 Zu beliebigem reellem $a > 0$ kann die Quadratwurzel $\sqrt{a}$ mit dem *divisionsfreien* Iterationsverfahren

$$x_{k+1} = \frac{1}{2} x_k (3 - a x_k^2) , \quad k = 0, 1, 2, \ldots$$

bei beliebigem Startwert $x_0 > 0$ bestimmt werden. Die Abbildung $g(x) = \frac{1}{2}x(3 - ax^2)$ besitzt den positiven Fixpunkt $x^* = \sqrt{1/a}$, aus dem man nach einer zusätzliche Multiplikation mit a den gesuchten Wert $\sqrt{a}$ erhält.

Mit der Ableitung $g'(x) = \frac{3}{2}(1 - ax^2)$ ergibt sich am Fixpunkt der Ableitungswert $\sigma = g'(x^*) = \frac{3}{2}(1 - a(x^*)^2) = 0$, weshalb Satz 1.6 von Ostrowski die lokale Konvergenz der Iterationswerte garantiert. Zudem ist nach Satz 3.1 diese Konvergenz mindestens Q-quadratisch. ◀

3.2 Das Newton-Verfahren

Ausgangspunkt sei das nichtlineare Gleichungssytem (2) in Nullstellenform

$$f(x) \; = \; 0 \qquad \text{mit} \quad f : \; D \subset \mathbb{R}^n \to \; \mathbb{R}^n \, , \; D \text{ offen}$$

mit dem Lösungsvektor x^* . Der Einfachheit halber werde vorausgesetzt, daß f auf D mindestens zweimal stetig differenzierbar ist. Nach Satz 1.1 läßt sich das Nullstellenproblem in eine Fixpunktaufgabe $x = g(x)$ mit

$$g(x) := x - \Phi(x) f(x)$$

überführen. Darin ist $\Phi : D \to \mathbb{R}^{n \times n}$ eine beliebige hinreichend oft stetig differenzierbare Matrixfunktion, die für jedes $x \in D$ regulär ist. Äquivalenzsatz 1.1 garantiert dann, daß jeder Fixpunkt von g auch eine Nullstelle von f ist und umgekehrt.

Wir betrachten deshalb die allgemeine Form der Iterationsverfahren

$$x^k \; = \; x^{k-1} - \Phi(x^{k-1}) f(x^{k-1}) \, , \quad k = 1, 2, 3, ... \tag{33}$$

mit Startvektor x^0 und Matrixfunktion $\Phi(x)$ und klären zuerst die Frage, wie die Matrix Φ zu wählen ist, um mindestens Q-quadratische Konvergenz garantieren zu können. Man bildet dazu die Jacobi-Matrix von $g(x)$ am Fixpunkt

$$G(x^*) = g'(x^*) = I - \Phi(x^*) f'(x^*) ,$$

wobei $f'(x^*) = 0$ zu beachten ist, und wendet Satz 3.1 an. Mit der Jacobimatrix von f

$$F(x) = f'(x) = (f_{ij}) = \left(\frac{\partial f_i}{\partial x_j}\right)$$

liefert die Bedingung $G(x^*) = O$ die folgende Darstellung

$$\Phi(x^*) = [F(x^*)]^{-1} = [f'(x^*)]^{-1} .$$

Da der Fixpunkt im allgemeinen nicht bekannt ist, läßt sich diese Gleichheit stets erreichen, wenn die Matrix für alle $x \in D$ zu

$$\Phi(x) := [F(x)]^{-1} = [f'(x)]^{-1}$$

gewählt wird, vorausgesetzt $F(x)$ ist regulär. Einsetzen in das allgemeine Verfahren 33 liefert dann das *Newton-Verfahren*

$$\boxed{x^{k+1} \;=\; x^k - [F(x^k)]^{-1} f(x^k) \;, \quad k = 0, 1, 2, 3, \ldots .} \tag{34}$$

Dieses einfache Einschrittverfahren, im eindimensionalen Falle auch als *Tangenten-Näherungsverfahren* bezeichnet, bildet eines der grundlegendsten und leistungsfähigsten mathematischen Näherungsverfahren für differenzierbare Abbildungen f. Folgende Aspekte sollen dies verdeutlichen:

1. Gemäß unserer Konstruktion der Iterationsabbildung

   $$g(x) := x - [F(x)]^{-1} f(x)$$

   ist $G(x^*) = O$ und infolgedessen das Verfahren unter entsprechenden Voraussetzungen *mindestens quadratisch konvergent*. Es gehört damit zu den schnellsten Basisverfahren für große Problemklassen. (Für eingeschränkte Problemklassen kann man Verfahren höherer Konvergenzordnung konstruieren, die jedoch meist nur im eindimensionalen Fall dem Newton-Verfahren in der Rechenzeit überlegen sind.)

2. Nach Satz 1.6 von Ostrowski gilt wegen $G(x^*) = O$ nun auch $\sigma = \varrho(G(x^*)) < 1$, so daß eine *lokale Konvergenz des Newton-Verfahrens* gegen eine vorausgesetzte Lösung x^* vorliegt, wenn g stetig differenzierbar in D ist. Diese Differenzierbarkeitseigenschaft setzt allerdings die Regularität der Jacobimatrix $F(x) = f'(x)$ auf dem gesamten Definitionsbereich D voraus. Ein Konvergenzsatz, der nur die Regularität von $f'(x^*)$ benötigt, wird im nächsten Abschnitt angegeben.

3. Entwickelt man $f(x)$ an der Stelle x^k in eine Taylorreihe

   $$f(x) \;=\; f(x^k) + f'(x^k)(x - x^k) + R(x, x^k) \,,$$

   so liefert die rechte Seite bei Vernachlässigung des Restgliedes $R(x, x^k)$ die *Linearisierung der Funktion f* am Näherungspunkt x^k. Anstelle der Gleichung $f(x) = 0$ löst man nun die linearisierte Gleichung mit der Lösung $x = x^{k+1}$

   $$f(x^k) \;+\; f'(x^k)(x^{k+1} - x^k) \;=\; 0 \,,$$

   und erhält nach Umstellung das Newton-Verfahren (34). Es stellt damit eine iterative Linearisierung der gegebenen Funktion an den Näherungswerten x^k dar.

Die Implementation des Newton-Verfahrens erfordert die Berechnung von $f(x^k)$ und der Jacobi-Matrix $F(x^k)$ sowie die Bestimmung der Näherung x^{k+1}. Nehmen wir an, die Jacobi-Matrix sei voll besetzt, und direkte Verfahren werden zur Lösung der linearen Gleichungssysteme eingesetzt. In jedem Schritt des Newton-Verfahrens ist dann die Invertierung der Jacobi-Matrix $F(x^k)$ erforderlich. Der beträchtliche arithmetische Aufwand läßt sich jedoch auf $\frac{1}{3}$ reduzieren, wenn man stattdessen in jedem Schritt ein lineares Gleichungssystem löst. Das erreicht man, wenn man die Verfahrensgleichung (34) nach der Newton-Korrektur $d^k = x^{k+1} - x^k$ umstellt und mit der Jacobi-Matrix $F(x^k)$ multipliziert. So entsteht die *praktikable Form des Newton-Verfahrens*:

Löse für $k = 0, 1, 2, ...$ das lineare Gleichungssystem

$$\begin{aligned} F(x^k)\, d^k &= -f(x^k)\,, \\ x^{k+1} &= x^k + d^k\,. \end{aligned} \tag{35}$$

Der *arithmetische Aufwand* pro Newton-Schritt besteht nun in der Lösung eines linearen Gleichungssystems mit n Unbekannten, wofür die LU-Zerlegung oder QR-Zerlegung eingesetzt werden kann. Dafür sind $O(n^3)$ Gleitpunktoperationen (floating point operations, flops) erforderlich. Allerdings sind die n Funktionswerte von $f(x^k)$ und die n^2 partiellen Ableitungen von $F(x^k)$ ebenfalls zu berechnen, was zu einem beträchtlichen *funktionellen Aufwand* des Newton-Verfahrens führen kann.

Der zugehörige Algorithmus `newton` erfordert als Input die Funktionen f und F, einen Startwert x und die (absolute und relative) Toleranz $tolabs, tolrel$.

Algorithmus 3.1 (Newton-Verfahren)

Function newton$(f, F, x, tolabs, tolrel)$

1. Berechne Toleranz $tol = tolrel \cdot ||f(x)|| + tolabs$
2. Do while $||f(x)|| > tol$
   1. Berechne Jacobi-Matrix $F(x)$
   2. Zerlege $F(x) = LU$
   3. Löse $LU \cdot d = -f(x)$
   4. $x = x + d$
   5. Berechne $f(x)$
3. Return x

Dieser Algorithmus läßt sich leicht im Matrix-orientierten Numerik-System MATLAB implementieren. Die angegebene Parameterliste $(f, F, x, tolabs, tolrel)$ sollte allerdings durch eine maximale Iterationszahl $maxit$ ergänzt werden, um in ungünstigen Fällen eine Endlosschleife zu vermeiden. Außer der erhaltenen Lösungsnäherung x liefert die MATLAB-Funktion

`newton` das Residuum $res = ||f(x)||$ in der Euklidischen Norm und die Anzahl *iter* der ausgeführten Iterationsschritte.

```
function [x,res,iter] = newton (fname,Dfname,x0,tolabs,tolrel,maxit);
% Algorithmus 3.1 : Newton-Verfahren
% zur Loesung nichtlinearer Gleichungssysteme
% ****************************************
%     [x,res,iter] = newton(fname, Dfname, x0, tolabs, tolrel, maxit)
%     benutzt das Newton-Verfahren,  um ausgehend von x0 einen Vektor
%     x zu finden, der das nichtlineare Gleichungssystem f(x) = 0 bis
%     auf eine Genauigkeit  tol = tolrel*||f(x)||+tolabs loest. Dabei
%     ist f die gegebene Modellfunktion.
%
%     Fuer die Ausfuehrung sind folgende Parameter notwendig:
%        fname   Name(Pfad) der Matlabfunktion zur Berechnung
%                der Funktionwerte f(x)
%        Dfname  Name(Pfad) der Matlabfunktion zur Berechnung
%                der ersten Ableitungen f'(x)
%        x0      Startwert (Spaltenvektor)
%        tolabs  absolute Fehlerschranke fuer das Residuum ||f(x)||
%        tolrel  relative Fehlerschranke fuer das Residuum ||f(x)||
%        maxit   maximale Anzahl der Iterationen
%
%     Ergebnisparameter sind:
%        x       Ergebnisvektor (Spaltenvektor)
%        res     Euklidische Norm des berechneten Wertes ||f(x)||
%        iter    Anzahl der ausgefuehrten Iterationen
%
iter = 0;  x = x0;
fx = feval(fname,x);
tol= tolrel * norm(fx) + tolabs;
% Iterationszyklus
while (norm(fx) > tol) & (iter < maxit)
    Dfx = feval(Dfname,x);
    d   = -Dfx \ fx;
    x   = x + d;
    fx  = feval(fname,x);
    iter= iter + 1;
end  % while
res = norm(fx);
% ****************************************************************
```

Im Euklidischen Raum $E = \mathbb{R}^{n+1} = \{(x_1, x_2, \ldots, x_n, y)\}$ kann das Newton-Verfahren auch geometrisch interpretiert werden. Jede gegebene Gleichung besitzt einen Funktionsgraphen $y = f_i(x_1, x_2, \ldots, x_n)$, der in E eine n-dimensionale Hyperfläche beschreibt. Deren Spur für $y = 0$ definiert in $\mathbb{R}^n$ eine $(n-1)-$dimensionale Lösungsmannigfaltigkeit M_i. Der Durch-

schnitt $M = \cap M_i$ dieser Mengen für $i = 1(1)n$ liefert schließlich die Lösungen x^*. Die Linearisierung von f an der Stelle x^k lautet nun

$$y = F(x^k)(x - x^k) \; + \; f(x^k)$$

bzw. in komponentenweiser Notation

$$y = \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(x_1^k, x_2^k, \ldots, x_n^k)(x_j - x_j^k) + f_i(x_1^k, x_2^k, \ldots, x_n^k) \; , \quad i = 1(1)n \; .$$

Sie beschreibt für jedes i genau die Tangential-Hyperebene an die Hyperfläche im Punkt $x^k = (x_1^k, x_2^k, \ldots, x_n^k, f_i(x_1^k, x_2^k, \ldots, x_n^k)) \in \mathbb{R}^{n+1}$. Die Gleichungen des Newton-Verfahrens geben demzufolge die $(n-1)-$dimensionalen Spuren dieser Hyperebenen in $\mathbb{R}^n$ an, deren Durchschnittsmenge den neuen Näherungspunkt x^{k+1} definiert. Der Begriff *Tangenten-Näherungsverfahren* ist somit auch im n-dimensionalen Fall zutreffend.

Beispiel 3.2 Das nichtlineare Gleichungssystem

$$\begin{array}{rcccl} f_1(x_1, x_2) & = & 1 - x_1^2 - x_2^2 & = & 0 \\ f_2(x_1, x_2) & = & -2x_1 + x_2 & = & 0 \end{array} \tag{36}$$

besitzt die zwei Lösungen $x^* = \pm(\frac{1}{5}\sqrt{5}, \frac{2}{5}\sqrt{5})$. Die Flächen $y = f_i(x_1, x_2)$ in $\mathbb{R}^3$ werden in Abb.1 dargestellt, während Abb. 2 die Spuren dieser Flächen in der x_1x_2-Ebene zeigt.

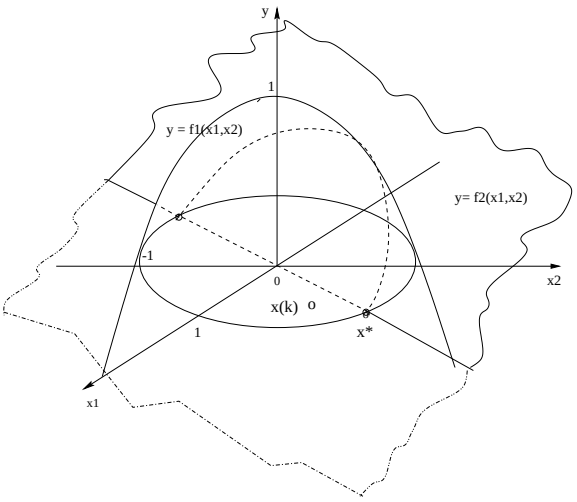

Abbildung 1:

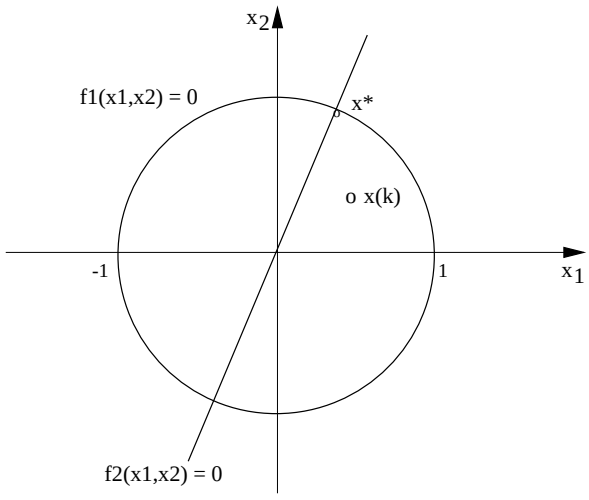

Abbildung 2:

In Abb. 3 wird die Tangentialebene 1 an die Fläche 1 im Punkt $P_k = (x_1^k, x_2^k, f_1(x_1^k, x_2^k))$ gestrichelt dargestellt, wogegen die Tangentialebene 2 identisch mit der ebenen Fläche 2 ist. Der neue Iterationspunkt x^{k+1} ergibt sich nun als Schnitt der Spuren dieser beiden Tangentialebenen in der x_1x_2-Ebene (vgl. auch Abb. 4).

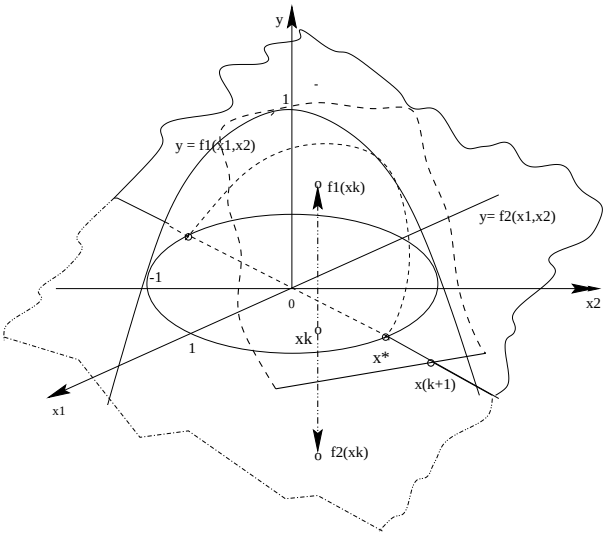

Abbildung 3:

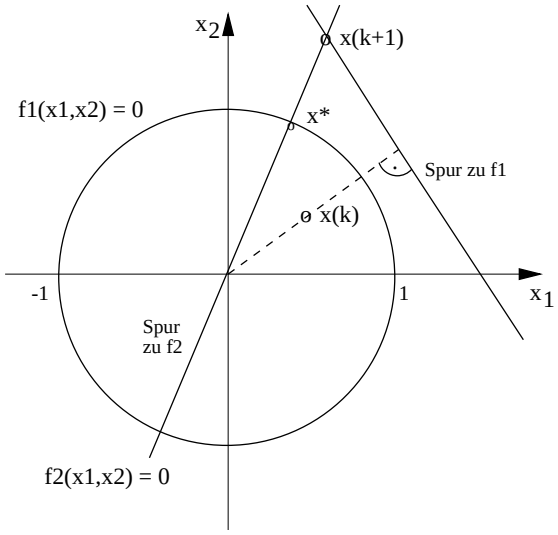

Abbildung 4:

Die Funktion f und ihre Jacobimatrix Df sind als M-Files bereitzustellen:

```
% Beispielsystem - Funktionen
function f = bsp10(v)
%
x = v(1); y = v(2);
f = zeros(2,1);
f(1) = 1 - x^2 - y^2;
f(2) = -2*x + y;
```

```
% Beispielsystem - Jacobimatrix
function Df = Dbsp10(v)
%
x = v(1); y = v(2);
Df = zeros(2,2);
Df(1,:) = [-2*x, -2*y];
Df(2,:) = [-2,    1  ];
```

Mit den zwei MATLAB-Aufrufen

```
[Loesung,Residuum,Iterationen] =
       newton('bsp10', 'Dbsp10', [3, 1.7]',1e-12,1e-12,15)
[Loesung,Residuum,Iterationen] =
       newton('bsp10', 'Dbsp10', [-3, -2]',1e-12,1e-12,15)
```

liefert das Newton-Verfahren mit 6 Iterationsschritten die gewünschten Lösungen

```
Loesung =
    4.472135955000063e-001
    8.944271910000127e-001
Residuum =
    2.163824674994430e-013
Iterationen =
     6
```

```
Loesung =
   -4.472135954999956e-001
   -8.944271909999912e-001
Residuum =
    1.685318551380988e-013
Iterationen =
     6
```

3.3 Reguläre Nullstellen, lokale und semilokale Konvergenz

Wir wollen nun die theoretische Frage behandeln, unter welchen Bedingungen das Newton-Verfahren gegen eine Nullstelle x^* konvergiert. Bezüglich der angenommenen Voraussetzun-

gen unterscheidet man dabei 3 Arten der Konvergenz, die lokale, semilokale und globale Konvergenz.

- Wenn voraussetzt wird, daß ein Fixpunkt x^* existiert und der Konvergenzsatz dann garantiert, daß „eine (Kugel-)Umgebung S dieses Fixpunktes existiert, so daß das Iterationsverfahren für alle $x^0 \in S$ gegen x^* konvergiert“, so heißt dieses Verfahren *lokal konvergent.* Über die Lage und Größe der Umgebung S ist in praxi meist nichts bekannt; man weiß nur, daß für hinreichend nahe am Fixpunkt liegende Starrtpunkte das Verfahren konvergieren wird. Ein typisches Beispiel dafür ist der Satz 1.6 von Ostrowski.

- Muß die Existenz eines Fixpunktes x^* nicht vorausgesetzt werden und können die Voraussetzungen eines Konvergenzsatzes für einen Startpunkt x^0 verifiziert werden, so heißt das betreffende Verfahren *semilokal konvergent.* Dazu ist häufig eine Menge D zu beschreiben, so daß für Startpunkte $x^0 \in D$ die Iteration konvergiert. Satz 1.2 liefert ein Beispiel für eine semilokale Konvergenzaussage.

- Ist $D \subset \mathbb{R}^n$ ein im allgemeinen großer *vorgegebener* Bereich (z.B. eine Kugel um 0, ein n-dimensionales endliches oder unendliches Intervall), so heißt ein Iterationsverfahren *global konvergent auf* D, falls es für jeden Startpunkt $x^0 \in D$ gegen einen Fixpunkt x^* in D konvergiert. Verfahren dieser Art werden in Teil 2 behandelt.

Wir wollen zuerst einen lokalen Konvergenzsatz für das Newton-Verfahren formulieren und beweisen. Dazu betrachten wir die Gleichung in Nullstellenform

$$f(x) \;=\; 0 \;, \quad f : D \subset \mathbb{R}^n \to \mathbb{R}^n \;, \quad D \text{ offen} \tag{37}$$

und setzen die Existenz einer regulären Lösung $x^* \in D$ voraus.

Definition 3.2 (Reguläre Lösung)
Eine Lösung $x^* \in D$ heißt regulär (isoliert)*, wenn*

- *eine Kugel $S^* = S(x^*, \delta^*) = \{x \mid ||x - x^*|| \le \delta^*\}$ um x^* mit $S^* \in int(D)$ existiert,*
- *die Jacobi-Matrix $F(x) = f'(x)$ auf S^* Lipschitz-stetig ist und*
- *die Jacobi-Matrix $F(x^*)$ regulär ist.*

Die Regularität der Nullstelle stellt in der Tat eine *Standardvoraussetzung* an das zu lösende Problem dar. Bei singulärer Jacobi-Matrix $F(x^*)$ ist das Verhalten des Newton-Verfahrens überaus kompliziert: Während in einigen Fällen noch eine lineare Konvergenz eintritt (z.B. bei skalaren Nullstellenproblemen), versagt das Verfahren in höherdimensionalen Systemen fast immer.

Eine Lösung x^* ist *geometrisch isoliert*, falls eine Umgebung S existiert, in der keine weitere Lösung $x^{**} \neq x^*$ liegt. Die Regularität einer Nullstelle darf deshalb nicht mit deren *geometrischer Isoliertheit* verwechselt werden.[1] Aus der Regularität folgt im übrigen stets die geometrische Isoliertheit einer Lösung.

Falls f zweimal stetig differenzierbar auf der gesamten Menge D ist, so ist offenbar die Bedingung $\det F(x^*) \neq 0$ hinreichend für die Regularität einer Lösung $x^* \in int(D)$.

[1] Der häufig anzutreffende Begriff „Isoliertheit“ wird deshalb hier nicht benutzt

Beispiel 3.3 Für das System mit zweimal stetig differenzierbarem $f \in \mathbb{R}^2$

$$\begin{aligned} f_1(x_1, x_2) &= (x_1 - x_2)^2 &= 0 \\ f_2(x_1, x_2) &= x_1 + x_2 - 2 &= 0\,. \end{aligned}$$

erhält man die Jacobi-Matrix

$$F(x) = \begin{pmatrix} 2(x_1 - x_2) & -2(x_1 - x_2) \\ 1 & 1 \end{pmatrix}.$$

An der Lösung $x^* = (1, 1)$ hat deren Determinante den Wert

$$\det F(x^*) = \begin{vmatrix} 0 & 0 \\ 1 & 1 \end{vmatrix} = 0\,,$$

weshalb x^* nicht regulär ist. Modifiziert man die erste Gleichung zu

$$\begin{aligned} f_1(x_1, x_2) &= (x_1 - x_2)^2 - 4 &= 0 \\ f_2(x_1, x_2) &= x_1 + x_2 - 2 &= 0\,, \end{aligned}$$

so ändert man die Jacobi-Matrix $F(x)$ damit nicht. Für die Lösung $x^* = (2, 0)$ besitzt sie nun die Determinante

$$\det F(x^*) = \begin{vmatrix} 4 & -4 \\ 1 & 1 \end{vmatrix} = 8 \neq 0\,,$$

also ist diese Lösung regulär.

Falls man das Newton-Verfahren auf das nicht-reguläre Problem anwendet, so wird man bei zufällig gewähltem Startpunkt x^0 i. allg. nur lineare Konvergenz erreichen. Mit den MATLAB-Funktionsaufrufen

```
[Loesung,Residuum,Iterationen] =
      newton('bsp11', 'Dbsp11', [3, 1.7]',1e-6,1e-6,10)
[Loesung,Residuum,Iterationen] =
      newton('bsp11', 'Dbsp11', [-3, -2]',1e-12,1e-12,30)
```

liefert MATLAB die Lösungsnäherungen mit beträchtlichem Iterationsaufwand

```
Loesung =
    1.000634765625000e+000
    9.993652343749999e-001
Residuum =
    1.611709594726450e-006
Iterationen =
    10
```

```
Loesung =
    9.999990463256836e-001
    1.000000953674316e+000
Residuum =
    3.637978807091713e-012
Iterationen =
    19
```

Die erreichte Lösungsgenauigkeit ist mit $\sqrt{tol}$ zudem gering. ◀

Für den folgenden lokalen Konvergenzsatz benötigen wir zwei grundlegende Hilfssätze, die vorab angegeben werden.

Lemma 3.1 (Störungslemma)
Seien $A, B \in \mathbb{R}^{n\times n}$ *Matrizen, wobei* A *regulär ist mit* $||A^{-1}|| \leq \alpha$ *. Ist desweiteren* $||A - B|| \leq \beta$ *mit* $\varkappa := \alpha\beta < 1$ *, so ist auch* B *regulär, und es gelten die Abschätzungen*

$$||B^{-1}|| \;\leq\; \frac{\alpha}{1-\varkappa} \tag{38}$$

$$||A^{-1} - B^{-1}|| \;\leq\; \frac{\alpha^2}{1-\varkappa}||A-B|| \;\leq\; \frac{\alpha^2\beta}{1-\varkappa}\,. \tag{39}$$

Beweis: Vgl. [13], S. 23. □

Lemma 3.2 *f sei auf der konvexen Menge $D_0 \subset \mathrm{int}\, D$ differenzierbar, und $f'(x)$ sei dort Lipschitz-stetig mit der Konstanten $L > 0$. Dann gilt für alle $x, y, z \in D_0$*

$$||f(y) - f(x) - f'(z)(y-x)|| \;\leq\; \frac{L}{2}||y-x||\,\{||y-z|| + ||x-z||\} \tag{40}$$

Beweis: Für $x, y, z \in D_0$ gilt nach dem Mittelwertsatz

$$\begin{aligned}
||f(y) - f(x) - f'(z)(y-x)|| &= \left\| \int_0^1 \{f'(ty + (1-t)x) - f'(z)\}\,(y-x)\,dt \right\| \\
&\leq \int_0^1 L||ty + (1-t)x - z|| \cdot ||y-x||\,dt \\
&\leq L\left\{ \int_0^1 t\,dt \cdot ||y-z|| \;+\; \int_0^1 (1-t)\,dt \cdot ||x-z|| \right\} \cdot ||y-x|| \\
&= \frac{L}{2}||y-x||\,\{||y-z|| + ||x-z||\}\;. \qquad \square
\end{aligned}$$

Mit Hilfe dieser Lemmata kann man die lokale Konvergenz des Newton-Verfahrens (34) für reguläre Nullstellen nachweisen:

Satz 3.2 (Lokale Konvergenz) *$f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ besitze die reguläre Lösung $x^* \in D$. Dann existieren Konstanten $\delta > 0$ und $Q > 0$ mit $Q\delta < 1$ und eine Kugel $S = S(x^*, \delta)$, so daß folgende Behauptungen gelten:*

(i) *Das Newton-Verfahren ist für jeden Startwert $x^0 \in S$* durchführbar *mit $x^k \in S$ für alle $k \geq 0$.*

(ii) $\lim_{k\to\infty} x^k = x^*$.

(iii) x^* *ist die* einzige *Lösung in der Kugel S .*

(iv) *Die Konvergenz ist mindestens* Q-quadratisch, *d.h.*

$$\|x^{k+1} - x^*\| \;\leq\; Q\|x^k - x^*\|^2 \quad \forall\, k \geq k_0\,.$$

BEWEIS: Wir zeigen zuerst Behauptung (i). Seien $\delta^* > 0$ der gegebene Radius von $S = S(x^*, \delta^*)$, $L > 0$ die vorausgesetzte Lipschitz-Konstante und $\alpha := ||f'(x^*)-1||$ die Norm der Inversen. Mit fest gewähltem $\varkappa \in (0,1)$ setze man den Radius $\delta := \min\{\delta^*, \frac{\varkappa}{\alpha L}\}$ der Kugel $S = S(x^*, \delta)$ an. In S gilt

$$||f'(x^*)^{-1}|| \cdot ||f'(x^k) - f'(x^*)|| \;\le\; \alpha L||x^k - x^*|| \;\le\; \alpha L\delta \;\le\; \varkappa \;<\; 1 \qquad \forall\, x^k \in S\,.$$

Damit sind die Voraussetzungen des Störungslemmas mit $A = f'(x^*)$, $B = f'(x^k)$ erfüllt, weshalb die Inverse $B^{-1} = f'(x^k)^{-1}$ existiert und

$$||f'(x^k)^{-1}|| \;\le\; \frac{\alpha}{1-\varkappa} \;=:\; M \qquad \forall\, x^k \in S$$

gilt. Also kann die Iterierte x^{k+1} gebildet werden. Angenommen, $x^k \in S$. Dann erhält man für x^{k+1}

$$\begin{aligned} ||x^{k+1} - x^*|| &= ||x^k - f'(x^k)^{-1}f(x^k) - x^*|| \\ &= ||-f'(x^k)^{-1}f(x^k) + f'(x^k)^{-1}f'(x^k)(x^k - x^*)|| \\ &\le ||f'(x^k)^{-1}|| \cdot ||f(x^k) - f(x^*) - f'(x^k)(x^k - x^*)|| \\ &\le M\frac{L}{2}||x^k - x^*||^2 \end{aligned}$$

gemäß Lemma 3.2 mit $x = x^*$, $y = z = x^k$, denn $x^*, x^k \in S \subset S^*$. Damit ist bereits Behauptung (iv) mit $Q := M\frac{L}{2}$ bewiesen. Weitere Abschätzung liefert nun

$$||x^{k+1} - x^*|| \;\le\; M\frac{L}{2}||x^k - x^*|| \cdot ||x^k - x^*|| \;\le\; Q\delta||x^k - x^*||\,.$$

Wenn also zusätzlich $\lambda := Q\delta < 1$ gefordert wird, so ist

$$||x^{k+1} - x^*|| \;\le\; \lambda||x^k - x^*|| \;\le\; \lambda\delta \;\le\; \delta\,,$$

also $x^{k+1} \in S$. Schränkt man also den Radius δ auf

$$\delta := \min\left\{\delta^*, \frac{\varkappa}{\alpha L}, \frac{\lambda}{Q}\right\}$$

ein, so ist die Folge für jeden Startvektor $x^0 \in S$ definiert, und alle x^k liegen in S.
Durch rekursives Einsetzen erhält man schließlich

$$||x^k - x^*|| \;\le\; \lambda||x^{k-1} - x^*|| \;\le\; \ldots \lambda^k||x^0 - x^*|| \;\le\; \lambda^k\delta\,,$$

woraus wegen $\lambda < 1$ Behauptung (ii) folgt.
Wäre $x^{**} \in S$, $x^{**} \neq x^*$, eine weitere Lösung in S. Wählt man nun $x^0 := x^{**}$ als Startvektor, so verbleibt wegen $f(x^{**}) = 0$ die Iteration bei diesem Punkt; nach Behauptung (ii) gilt jedoch $x^* = \lim_{k\to\infty} x^k = x^{**}$, Widerspruch. Damit ist auch (iii) bewiesen. □

Da sich die Regularitätsvoraussetzung auf eine im allgemeinen unbekannte Lösung x^* bezieht, ist sie in der Praxis meist nicht überprüfbar. Geeigneter sind semilokale Sätze, die entsprechende Voraussetzungen von der *Startlösung* x^0 fordern. Wir geben den bekanntesten Satz - allerdings ohne den aufwendigen Beweis - an.

Satz 3.3 (L.V.Kantorovics)
Für die Startlösung x^0 des Newton-Verfahrens seien folgende Voraussetzungen *erfüllt:*

(i) *Es existiert die Inverse $F(x^0)^{-1}$ mit $\|F(x^0)^{-1}\| \leq \alpha$.*

(ii) *Für die Differenz der ersten beiden Näherungen gelte*

$$\|x^1 - x^0\| \;=\; \|F(x^0)^{-1} f(x^0)\| \leq \beta \; .$$

(iii) *f sei 2-mal stetig differenzierbar mit*

$$\sum_{k=1}^{n} \left| \frac{\partial^2 f_i(x)}{\partial x_j \partial x_k} \right| \;\leq\; \frac{\gamma}{n} \qquad , \quad i,j = 1(1)n \; ,$$

für alle $x \in S = \{x \mid \|x - x^0\| \leq 2\beta\}$.

(iv) $\alpha\beta\gamma \;\leq\; 1/2$.

Dann gelten folgende Behauptungen*:*

(i) *Das Newton-Verfahren ist für jeden Startwert $x^0 \in S$ unbeschränkt* durchführbar *mit $x^k \in S$ für alle $k \geq 0$.*

(ii) *Die Folge $\{x^k\}$ konvergiert gegen ein x^* mit $f(x^*) = 0$.*

(iii) *Für den Fehler gilt*

$$\|x^k - x^*\| \;\leq\; \frac{2\beta}{2^k} \qquad k = 0, 1, 2, \ldots \; .$$

Beweis: Vgl. [6], S.122-125. □

Falls in Voraussetzung (iv) sogar $\alpha\beta\gamma \;<\; 1/2$ gilt, so kann bewiesen werden, daß die Folge $\{x^k\}$ mindestens Q-quadratisch konvergiert.

4 Zwei Modifikationen des Newton-Verfahrens

4.1 Vereinfachtes Newton-Verfahren

Für großdimensionale Systeme ist der Aufwand pro Iterationsschritt auch in der praktikablen Form 35 des Newton-Verfahrens zu hoch. Man approximiert die Jacobi-Matrix $F(x^k)$ dann durch eine „geeignete“ konstante Matrix A. Im einfachsten Fall wählt man $A = F(x^0)$ und erhält so das *vereinfachte Newton-Verfahren (Sehnenverfahren, chord method)*:

(1) Berechne die Matrix $A = F(x^0)$ und deren LU-Zerlegung.
(2) Löse für $k = 0, 1, 2, ...$ das lineare Gleichungssystem

$$\begin{aligned} A\, d^k &= -f(x^k) \; , \\ x^{k+1} &= x^k + d^k \; . \end{aligned} \tag{41}$$

Offenbar ist nun der Iterationsschritt sehr effizient, denn der arithmetische Aufwand für eine Vorwärts-Rückwärtselimination ist nur von der Ordnung $O(n^2)$, und es sind lediglich n Funktionswerte in $f(x^k)$ zu berechnen. Dem steht eine langsamere Konvergenz entgegen. Denn die Iterationsfunktion g lautet nun $g(x) = x - A^{-1}f(x)$, womit sich die Ableitung

$$G(x^*) = I - A^{-1}F(x^*)$$

ergibt. Diese ist im allgemeinen verschieden von der Nullmatrix, weshalb für das Sehnenverfahren nur lineare Konvergenz erwartet werden kann.

Um diese Konvegenz des Verfahrens nachzuweisen, wird wie im Falle des Newton-Verfahrens die Regularität der Nullstelle x^* vorausgesetzt. Der Beweis verläuft ähnlich wie beim Newton-Verfahren und soll deshalb hier entfallen.

Satz 4.1 (Lokale Konvergenz) *$f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ besitze die reguläre Nullstelle $x^* \in D$. Dann existieren Konstanten $\delta > 0$ und $Q_1 > 0$, so daß folgende Behauptungen gelten:*

(i) *Das Sehnen-Verfahren ist für jeden Startwert $x^0 \in S = S(x^*, \delta)$* durchführbar *mit $x^k \in S$ für alle $k \geq 0$.*

(ii) $\lim_{k\to\infty} x^k = x^*$.

(iv) *Die Konvergenz ist* Q-linear *mit der Fehlerschätzung*

$$\|x^{k+1} - x^*\| \leq Q_1\|x^0 - x^*\|\|x^k - x^*\| , \quad k = 0, 1, 2, \ldots . \tag{42}$$

Beweis: Vgl. [7], S.76. □

Der Algorithmus `newton0` erfordert als Input die Funktionen f und F, einen Startwert x und die (absolute und relative) Toleranz $tolabs, tolrel$.

Algorithmus 4.1 (Sehnen-Verfahren)

Function newton0$(f, F, x, tolabs, tolrel)$

1. Berechne Toleranz $tol = tolrel \cdot ||f(x)|| + tolabs$
2. Berechne Jacobi-Matrix $F(x)$
3. Zerlege $F(x) = LU$
4. Do while $||f(x)|| > tol$
   1. Löse $LU \cdot d = -f(x)$
   2. $x = x + d$
   3. Berechne $f(x)$
5. Return x

Auch diesen Algorithmus kann man leicht als MATLAB-Funktion implementieren. Die angegebene Parameterliste sollte hier ebenfalls um eine maximale Iterationszahl *maxit* ergänzt werden. Damit lauten die Anweisungen dieser Funktion

```
function [x,res,iter] = newton0 (fname,Dfname,
                     x0, tolabs, tolrel,maxit);
% Algorithmus 4.1 : Sehnen-Verfahren
%
iter = 0;  x = x0;
fx   = feval(fname,x);
tol  = tolrel * norm(fx) + tolabs;
Dfx  = feval(Dfname,x);
% Iterationszyklus
while (norm(fx) > tol) & (iter < maxit)
    d   = -Dfx \ fx;
    x   = x + d;
    fx  = feval(fname,x);
    iter= iter + 1;
end  % while
res = norm(fx);
% ****************************************
```

Beispiel 4.1 Man bestimme in der Umgebung des Vektors $(1.2, 1, 7)$ eine Lösung $x^* = (x_1^*, x_2^*)$ des Gleichungssystems

$$\begin{aligned} f_1(x_1, x_2) &= 2x_1^3 - x_2^2 - 1 &= 0 \\ f_2(x_1, x_2) &= x_1 x_2^3 - x_2 - 4 &= 0\,. \end{aligned}$$

Die Jacobi-Matrix

$$F(x) = \begin{pmatrix} 6x_1^2 & -2x_2 \\ x_2^3 & 3x_1x_2^2 - 1 \end{pmatrix}$$

hat an der Startlösung $x^0 = (1.2, 1.7)$ den Wert

$$F(x^0) = \begin{pmatrix} 8.640 & -3.400 \\ 4.913 & 9.404 \end{pmatrix}.$$

Das Sehnenverfahren liefert folgende Iterationswerte und Residuen:

k	x_1^k	x_2^k	f_1^k	f_2^k
0	1.2	1.7		
1	1.234876263	1.721767620	-1.983E-1	+5.812E-1
2	1.233740351	1.660554527	-1.651E-3	-1.140E-2
3	1.234294695	1.661477530	+3.474E-4	-3.586E-4
4	1.234273794	1.661526582	-6.689E-6	-2.105E-6
5	1.234274509	1.661526432	+3.465E-7	-2.060E-7
6	1.234274483	1.661526467	-1.031E-8	+2.050E-9
7	1.234274484	1.661526467	+2.9E-10	-1.4E-10
8	1.234274484	1.661526467	+2.9E-10	-1.4E-10

Die Näherungslösung lautet dann $x_1^* = 1.234274484$, $x_2^* = 1.661526467$. Eine Fehleranalyse bestätigt die lineare Konvergenz des Sehnenverfahrens, allerdings mit einem kleinen Konvergenzfakor.

Vergleichen wir das Konvergenzverhalten mit dem des Newton-Verfahrens, indem wir die MATLAB-Funktionen `newton` und `newton0` mit verschiedenen Startpunkten und Genauigkeiten aufrufen (hier mit `newton` notiert).

```
[Loesung,Residuum,Iterationen] =
    newton('bsp12', 'Dbsp12', [1.2, 1.7]',1e-6,1e-6,10)
[Loesung,Residuum,Iterationen] =
    newton('bsp12', 'Dbsp12', [1.2, 1.7]',1e-15,1e-15,15)
[Loesung,Residuum,Iterationen] =
    newton('bsp12', 'Dbsp12', [30, 20]',1e-12,1e-12,25)
```

Mit diesen Aufrufen liefern die beiden Verfahren folgende Resultate:

```
Newton-Verfahren
================
Loesung =
    1.234274484114495e+000
    1.661526466795916e+000
Residuum =
    2.456182707821288e-013
Iterationen =
     3

Loesung =
    1.234274484114476e+000
    1.661526466795934e+000
Residuum =
    1.256073966947020e-015
Iterationen =
     5

Loesung =
    1.234274484361894e+000
    1.661526467417521e+000
Residuum =
    6.870128316322310e-009
Iterationen =
    13
```

```
Sehnen-Verfahren
================
Loesung =
    1.234274456463722e+000
    1.661526493839341e+000
Residuum =
    3.638760250363798e-007
Iterationen =
     4

Loesung =
    1.234274484114476e+000
    1.661526466795934e+000
Residuum =
    1.256073966947020e-015
Iterationen =
    10

Loesung =
    5.206548356832546e+000
    6.855395078378074e+000
Residuum =
    1.682972892866862e+003
Iterationen =
    50
```

Bei hinreichend nahe an der Lösung liegenden Startwerten ist offenbar das Sehnenverfahren trotz größerer Iterationszahlen vorzuziehen. Sind dagegen die Startwerte wie im 3. Aufruf wenig geeignet, so führt das Sehnenverfahren häufig nicht zum Erfolg. ◄

4.2 Shamanskii-Verfahren

Falls sich die Jacobimatrix $F(x^k)$ von Iteration zu Iteration nur wenig ändert, empfehlen sich mehrere Zwischenschritte mit ein- und derselben Matrix A. Dafür ist jedesmal nur eine LU-Zerlegung erforderlich. Man erhält eine Mischung aus dem Newton- und dem Sehnenverfahren, die häufig als *Shamanskii-Verfahren* bezeichnet wird. Es läßt sich als eine Variante des Newton-Verfahrens mit m inneren Iterationsschritten interpretieren:

Für $k = 0, 1, 2, \ldots$ iteriere

$$\begin{aligned} y^0 &= x^k - [F(x^k)]^{-1} f(x^k)\,, \\ y^{j+1} &= y^j - [F(x^k)]^{-1} f(y^j)\,, \quad j = 0(1)m-1 \\ x^{k+1} &= y^m\,. \end{aligned} \tag{43}$$

Das Newton-Verfahren liegt offenbar in dem Sonderfall $m = 0$ vor, während im Grenzfall $m \to \infty$ dieses Verfahren in das Sehnen-Verfahren übergeht. Nimmt man die Werte y^j der inneren Iteration als Hilfsgrößen an und betrachtet lediglich die Folge $\{x^k\}$ der äußeren Iterierten, so kann man unter geeigneten Voraussetzungen nachweisen, daß diese Folge mindestens mit Q-Ordnung $m+2$ gegen die Lösung x^* konvergiert. So kann mit relativ geringem Zusatzaufwand (1 Berechnung von f und eine Vorwärts-Rückwärtselimination) eine höhere Konvergenzordnung als beim gewöhnlichen Newton-Verfahren erreicht werden.

Der Algorithmus `shamanskii` erfordert als Input ebenfalls die Funktionen f und F, einen Startwert x und die (absolute und relative) Toleranz $tolabs, tolrel$. Hinzu kommt die Anzahl m der inneren Iterationen.

Algorithmus 4.2 (Shamanskii-Verfahren)

Function shamanskii$(f, F, x, tolabs, tolrel, m)$

1. Berechne Toleranz $tol = tolrel \cdot ||f(x)|| + tolabs$
2. Do while $||f(x)|| > tol$
   1. Berechne Jacobi-Matrix $F(x)$
   2. Zerlege $F(x) = LU$
   3. For $j = 0(1)m$
      1. Löse $LU \cdot d = -f(x)$
      2. $x = x + d$
      3. Berechne $f(x)$
      4. If $||f(x)|| < tol$ Return x
3. Return x

Sämtliche Iterationswerte können in ein- und derselben Variablen x gespeichert werden. Wird die geforderte Genauigkeit bereits in einer inneren Iteration erreicht, so kann der entsprechende Iterationswert zurückgegeben werden.

Implementiert man diesen Algorithmus als MATLAB-Funktion, so sollte die Anzahl m der inneren Iterationen in die Parameterliste $(f, F, x, tolabs, tolrel, m)$ aufgenommen werden.

```
function [x,res,iter] = shamanskii(fname,
        Dfname,x0,tolabs,tolrel,m,maxit);
% Algorithmus 4.2 :  Shamanskii-Verfahren
% ************************************
% m - Anzahl der inneren Iterationen
%
iter = 0;  x = x0;
fx = feval(fname,x);
tol= tolrel * norm(fx) + tolabs;
% Aeusserer Iterationszyklus
while (norm(fx) > tol) & (iter < maxit)
    Dfx = feval(Dfname,x);
    % Innerer Iterationszyklus
    for j = 0 : m,
        d   = -Dfx \ fx;
        x   = x + d;
        fx  = feval(fname,x);
        res = norm(fx);
        iter= iter + 1;
        if res < tol
            return
        end   % if
    end   % for
end  % while
% ************************************
```

Die lokale Konvergenz des Shamanskii-Verfahrens läßt sich mit dem Konvergenzsatz 4.1 des Sehnenverfahrens (vereinfachten Newton-Verfahrens) verifizieren. Damit wird allerdings auch die Regularität der Nullstelle x^* vorausgesetzt.

Satz 4.2 (Shamanskii-Verfahren) *$f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ besitze die reguläre Nullstelle $x^* \in D$ und es sei $m \geq 0$. Dann existieren Konstanten $\delta > 0$ und $Q_2 > 0$, so daß folgende Behauptungen gelten:*

(i) Das Shamanskii-Verfahren ist für jeden Startwert $x^0 \in S = S(x^*, \delta)$ durchführbar *mit $x^k \in S$ für alle $k \geq 0$.*

(ii) Es gilt $\lim_{k\to\infty} x^k = x^*$ Q-überlinear.

(iv) Die Konvergenz ist mindestens von Q-Ordnung $m + 2$ *mit der Fehlerschätzung*

$$\|x^{k+1} - x^*\| \;\leq\; Q_2 \|x^k - x^*\|^{m+2} , \quad k \geq k_0 . \tag{44}$$

BEWEIS: Wir betrachten für einen Index k die Folge $\{y^j\}$ der inneren Iteration. Wegen der Regularitätsvoraussetzungen garantiert Satz 4.1 die Konstanten $Q_1 > 0$ und $\delta > 0$, mit denen aus $x^k \in S(x^*, \delta)$ folgt, daß alle inneren Iterierten y^j auch in dieser Kugel liegen. Damit ist auch $x^{k+1} \in S$. Darüber hinaus schätzt man mit (42) rekursiv ab

$$\begin{aligned}
\|y^j - x^*\| &\leq Q_1\|x^k - x^*\|\|y^{j-1} - x^*\| \\
&\leq Q_1^2\|x^k - x^*\|^2\|y^{j-2} - x^*\| \\
&\leq \cdots\cdots\cdots\cdots\cdots\cdots \\
&\leq Q_1^j\|x^k - x^*\|^j\|y^0 - x^*\| \\
&\leq Q_1^{j+1}\|x^k - x^*\|^{j+2} .
\end{aligned}$$

Für $j = m$ erhält man die Fehlerschätzung (44) mit der Konstanten $Q_2 := Q_1^{m+1}$. Mit hinreichend kleinem δ liefert (44) schließlich die Konvergenzaussage (ii). □

Beispiel 3.2 Wir wenden das Shamanskii-Verfahren mit den inneren Iterationszahlen $m = 1, 2, 5, 10$ auf folgende Parametersätze an:

```
[Loesung,Residuum,Iterationen] =
    shamanskii('bsp10', 'Dbsp10', [3, 1.7]',1e-6,1e-6,m,10)
[Loesung,Residuum,Iterationen] =
    shamanskii('bsp10', 'Dbsp10', [3, 1.7]',1e-12,1e-12,m,15)
[Loesung,Residuum,Iterationen] =
    shamanskii('bsp10', 'Dbsp10', [-3, -2]',1e-12,1e-12,m,15)
```

In der folgenden Tabelle wird die Anzahl k aller Iterationen des Newton-, des Shamanskii- und des Sehnen-Verfahrens gegenübergestellt.

tol	Newton-Verfahren	Shamanskii-Verfahren $m=1$	$m=2$	$m=5$	$m=10$	Sehnen-Verfahren
10^{-6}	5	7	7	10	11*	28*
10^{-12}	6	8	9	13	16	50*
10^{-12}	6	8	10	13	16	50*

In den mit $*$ markierten Fällen wurde die geforderte Genauigkeit *tol* nicht vollständig erreicht. Beachtet man jedoch, daß das Newton-Verfahren in dritten Aufruf $k = 6$ Berechnungen der Jacobi-Matrix und einen arithmetischen Aufwand von $k \cdot \frac{2}{3}n^3$ erfordert, so liefert das Shamanskii-Verfahren die Lösung bereits für $k = 16$, $m = 10$ mit

$$\left[\frac{k-1}{m+1}\right] + 1 = 2 \quad \text{Jacobi-Matrizen}$$

und einem arithmetischen Aufwand von $2 \cdot \frac{2}{3}n^3$. ◀

Der Vorteil dieser Verfahren tritt bei großdimensionalen Gleichungssystemen stärker hervor. Das Verfahren kann zudem verbessert werden, wenn die Anzahl m ergebnisabhängig gewählt und gegebenenfalls adaptiv geändert wird. Das kann durch Kontrolle der Fehlerentwicklung der inneren Iteration erfolgen (vgl. [7]).

Literatur

[1] Adams, E.; Kulisch, U. (Hrsg.): *Scientific Computing with Automatic Result Verification.* Academic Press, San Diego 1993.

[2] Allgower, E. L.; Georg, K.: *Numerical Continuation Methods.* Springer – Verlag, Berlin 1990.

[3] Deuflhard, P.; Hohmann, A.: *Numerische Mathematik I.* 2. Auflage, W. de Gruyter, Berlin 1993.

[4] Gramlich, G.; Werner, W.: *Numerische Mathematik mit MATLAB.* dpunkt.verlag GmBH, Heidelberg 2000.

[5] Hackbusch, W.: *Iterative Lösung großer schwachbesetzter Gleichungssysteme.* B.G.Teubner, Stuttgart 1991.

[6] Isaacson, E.; Keller, H. B.: *Analyse numerischer Verfahren.* Verlag Harry Deutsch, Frankfurt 1972.

[7] Kelley, C.T.: *Iterative Methods for Linear and Nonlinear Equations.* SIAM, Philadelphia 1995.

[8] Kosmol, P.: *Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben.* B.G.Teubner, Stuttgart 1993.

[9] Maess, G.: *Vorlesungen über numerische Mathematik. Band 1 und 2.* Akademie – Verlag, Berlin 1984.

[10] Ortega, J.M.; Rheinboldt, W.C.: *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York 1970.

[11] Rheinboldt, W.C.: *Methods for Solving Systems of Nonlinear Equations.* 4th ed., SIAM Publications, Philadelphia 1994.

[12] Saad, Y.: *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston 1995.

[13] Schwetlick, H.: *Numerische Lösung nichtlinearer Gleichungen.* Deutscher Verlag der Wissenschaften, Berlin 1979.

[14] Törnig, W.; Spellucci, P.: *Numerische Mathematik für Ingenieure und Physiker. Band 1 und 2.* 2. Auflage, Springer-Verlag, Berlin 1988.

[15] Trefethen, L.N.; Bau, D.: *Numerical Linear Algebra.* SIAM, Philadelphia 1997.